REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

14

1. REPORT NUMBER
UTEC-CSc-77-202

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER
9

4. TITLE (and Subtitle)
6
"Noise Suppression Methods for Robust Speech Processing".

5. TYPE OF REPORT & PERIOD COVERED
Semi-Annual Technical Tech. Rept.,
1 April 1977-30 Sept 1977,

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
10
Dr. Steven F. Boll

8. CONTRACT OR GRANT NUMBER(s)
7
N00143-77-C-0041

9. PERFORMING ORGANIZATION NAME AND ADDRESS
University of Utah
Computer Science Department
Salt Lake City, Utah 84112

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
Project: 76-RPA-3301

11. CONTROLLING OFFICE NAME AND ADDRESS
Defense Advanced Research Project Agency (DoD)
1400 Wilson Blvd.
Washington, D. C. 22209

11

12. REPORT DATE
October 1977

13. NUMBER OF PAGES
105

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
Naval Research Laboratory
4555 Overlook Avenue, S. W.
Mail Code 2415-A.M.

12 127p.

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
This document has been approved for public release and sale; its distribution is unlimited.

15
N00173-77-C-0041,
ARPA Order-3301

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Digital noise suppression; Linear Predictive Coding; Narrow band coded speech; Adaptive noise cancellation; Wiener filtering; Power spectrum; Autocorrelation, Spectral Averaging for Bias Estimation and Removal (SABER), Multirate Coding, CVSD, Widrow-Hoff LMS Algorithm, Pole-zero modeling and Estimation.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Robust speech processing in practical operating environments requires effective environmental and processor noise suppression. This report describes the technical findings and accomplishments during this reporting period for the research program funded to develop real time, compressed speech analysis-synthesis algorithms whose performance is invariant under signal contamination. Fulfillment of this requirement is necessary to insure reliable secure compressed speech transmission within realistic

→ next page

404 949

20. ABSTRACT con't.

military command and control environments. Overall contributions resulting from this research program include the understanding of how environmental noise degrades narrow band, coded speech, development of appropriate real time noise suppression algorithms, and development of speech paramenter identification methods that consider signal contamination as a fundamental element in the estimation process. This report describes the current research and results in the areas noise suppression using the SABER algorithm, dual input adaptive noise cancellation using the LMS algorithm, pole-zero parameter estimation and multirate coding and CVSD symulation.

TABLE OF CONTENTS

Page

I. D D Form 1473

II. REPORT SUMMARY

Section I Summary of Program for Reporting Period

III. RESEARCH ACTIVITIES

Section I Application of the SABER Method for Improved Spectral Analysis of Noisy Speech. Steven F. Boll 1

Section II Current Results on Dual Input Nonstationary Noise Suppression using LMS Adaptive Noise Cancellation. Dennis Pulsipher 50

Section III Estimation of the Parameters of an Autoregressive-Moving Average Process in the Presence of Noise. William Done 54

Section IV Multirate Signal Processing H. Ravindra 88

Section V Simulation of Continuously Variable Slope Delta Modulation H. Ravindra 101

Section I

Summary of Program for

Reporting Period

Program Objectives

To develop practical, real time methods for suppressing noise which has been acoustically added to speech.

To demonstrate that through the incorporation of the noise suppression methods, speech can be effectively analyzed for narrow band digital transmission in practical operating environments.

Summary of Tasks and Results

Introduction

This semi-annual technical report describes the current status in five research areas for the period 1 April 1977 through 30 September 1977.

Application of the SABER method for Improved Spectral

Analysis of Noisy Speech-Steven F. Boll.

A method is developed for reducing the effect of acoustically added background noise when spectrally analyzing speech using Linear Prediction. Fundamental to the method is the result that the spectral magnitude of speech plus noise can be modeled as the sum of magnitudes of speech and noise. This phase independent model allows for noise to be suppressed by subtracting the expected noise spectrum from the locally averaged speech spectrum. Using the Spectral Averaging for Bias Estimation and Removal, or SABER method, a noise reduction and corresponding signal to noise improvement of 15 dB is realized on both digitally added white Gaussian noise and acoustically added helicopter noise.

Current Results on Dual Input Nonstationary Noise Suppression Using LMS Adaptive Noise Cancellation-Dennis Pulsipher.

The previous Semi-Annual technical report described the successful application of the two microphone Widrow-Hoff Least Mean Square (LMS) algorithm for removing digitally added noise from speech. The method is now being applied for removing nonstationary acoustically added noise.

Preliminary results show that narrow band periodic noise can be completely eliminated and broad band colored noise can be reduced by at least 10 dB. When used in realistic operating environments, a filter length on the order of 300 ms. is required for broad band noise reduction.

Estimation of the Parameters of an Autoregressive-Moving Average Process in the Present of Noise-William Done.

This task considers an approach to parameter estimation in the presence of noise which involves the construction of a new model which explicity accounts for the effects of noise.

An analysis method is developed for parameter extraction which is significantly change from the standard LPC methods used when no noise is present. It is shown that the addition of noise to an all-pole or autoregressive (AR) process results in pole-zero or autoregressive-moving average ARMA process. Methods for estimating the parameters of the ARMA process are considered.

Multirate Signal Processing-H. Ravindra

The aim of this project is to simulate a system on a digital computer, which can increase or decrease the sampling rate of a digitized acoustic signal. These two operations are called Interpolation and Decimation respectively. This project is the first phase of a larger project which involves the simulation of a CVSD system, with an idea of studying the problems of tandeming. Since the CVSD performance (signal-to-noise ratio) is better at higher sampling rates, an Interpolation/Decimation scheme is required to translate the sampling rate from 6.67 KHz to higher rates.

Simulation of Continuously Variable Slope Delta Modulation (CVSD) H. Ravindra

A FORTRAN CVSD simulation was developed and implemented using the specification defined by Joe Tieney in Network Speech Compression note 15, April 23, 1974. Using the Multirate Signal Processing program CVSD coded speech can be generated at rates of 9.6, 16, 20, and 32 KBPS based on input speech sampled at 6.67 KHZ.

Future Efforts

SABER Development: The SABER algorithm will be modified to work in a "stand-alone" mode. In this implementation the speech will be windowed and transformed. The spectral magnitudes will be averaged and the noise bias removed. Then using the saved phase a time signal will be regenerated. This implementation will allow for noise suppression without affecting the bandwidth compression analyzer. Also intelligibility and quality measurements using the DRT will be conducted on the processed speech and compared with scores having no noise suppression.

Adaptive Noise Cancelling: Fundemental performance limits for the method's ability to reduce acoustically added noise in realistic environment will be established. Performance of the method in nonstationary noise environments will be demonstrated. Requirements for real time, practical implementation will be specified.

Parameter Estimation in Noise: Research will continue towards development of effective parameter extraction methods with consider noise as a fundamental component in the modeling process.

APPLICATION OF THE SABER METHOD FOR IMPROVED SPECTRAL ANALYSIS OF NOISY SPEECH

Steven F. Boll, Ph. D.

Chapter I

Abstract

A method is developed for reducing the effect of acoustically added background noise when spectrally analyzing speech using Linear Prediction. Fundamental to the method is the result that the spectral magnitude of speech plus noise can be modeled as the sum of magnitudes of speech and noise. This phase independent model allows for noise to be suppressed by subtracting the expected noise spectrum from the locally averaged speech spectrum. Using the Spectral Averaging for Bias Estimation and Removal, or SABER method, a noise reduction and corresponding signal-to-noise improvement of 15dB is realized on both digitally added white Gaussian noise and acoustically added helicopter noise.

Chapter II

Summary

This report describes an integrated noise suppression-speech analysis method for reducing the effect of background noise when spectrally analyzing speech using Linear Prediction. Basically it is shown that additive noise exhibits itself as a bias added to the desired speech spectrum. Through spectral averaging this bias will build up allowing it to be effectively removed using its expected value calculated during non-speech activity. The method is called SABER, an acronym for Spectral Averaging for Bias Estimation and Removal. This chapter summarizes the Objectives, Assumptions, Approach, and Results for the method. Detailed developments are provided in subsequent chapters.

Objectives

1. Develop and integrate a noise suppression algorithm into the narrow band LPC speech analysis algorithm.
2. Insure that the algorithm's effectiveness should be independent of any specific environment's noise characteristics.
3. Require the algorithm to only need a single microphone.

4. In implementing the algorithm, a minimal impact should result on existing narrow band systems.

   e.g. The same channel parameters should be used, thus allowing for the new system to be compatible with other LPC terminals.

5. In the absence of noise the method should generate synthetic speech equivalent in intelligibility and quality to standard LPC systems.

6. The method should not only be able to improve spectral resolution but also improve pitch and voicing estimation.

7. The method should use standard, well understood estimation techniques and be implementable in real time.

Assumptions

1. The background noise is acoustically or electrically added to the speech.

2. The background noise environment remains locally stationary to the degree that its spectral magnitude expected value just prior to speech activity equals its expected value during speech activity.

3. If the environment changes to a new stationary state, there exists enough time (on the order of 300 ms) to estimate a new background noise spectral magnitude expected value before speech activity commences.

4. If the environment changes to a new stationary state, the algorithm requires a speech activity detector to signal the program that speech has ceased and a new noise bias can be estimated.

Approach

1. The fundamental property is developed which demonstrates that the spectral magnitude of the noisy speech can be effectively modeled as the sum of magnitudes of speech and noise. This result is called the phase independent model.

2. Based upon the phase independent model, an estimate of the speech magnitude spectrum is calculated as follows:

   a. The noisy speech magnitude is averaged over stationary vocal tract intervals. Averaging yields a low variance bias noise term added to the speech spectrum.

   b. This sample mean is then removed by subtracting the expected noise spectrum from the averaged speech spectrum.

   c. Negative magnitude frequency components are then removed to further increase noise rejection.

   d. The resulting magnitude spectrum is then squared and inverse transformed yielding autocorrelations.

   e. From these autocorrelations predictor and reflection coefficients are calculated using the Levinson's recursion.

3. In addition, pitch and voicing information are calculated from the noise cancelled spectrum using a cepstral pitch tracker.

Results

1. The method will suppress white noise up to the limit allowed through reduction in variance of the sample mean. As shown in Section VI, the expected value of noise reduction equals approximate 15dB.
2. It is shown that the variance between actual speech spectrum and the SABER estimate equals the variance of the sample mean of the additive noise. This coupled with the fact that a 15dB reduction in noise energy is achievable suggested the following experiment for measuring signal-to-noise improvement. Speech having an average SNR of 25dB was processed with linear prediction and compared with speech having a SNR of 10dB and processed with SABER. Both informal listening tests and spectral comparisons demonstrate that the two outputs are essentially equivalent. This experiment suggests that a 15dB improvement in SNR is possible using SABER.
3. Preliminary experiments demonstrate that correct pitch values can be recovered when applying a cepstral pitch tracker to the noise cancelled SABER spectral estimate.

Chapter III

System Description

Introduction

This chapter describes the various algorithm stages for implementing the SABER method. Theoretical justification in support of these procedures is provided in the subsequent chapters.

Data Buffering

Data from the A/D converter is stored in a buffer system which is similar to standard vocoder designs. The analysis frame length should be at least twice as large as the maximum expected pitch period for adequate frequency resolution [1]. The analysis frames are advanced in time by overlapping by one-half the window length. As is shown in Appendix A, the one-half overlap is optimum for minimizing the variance of the sample mean of the magnitude spectra.

Spectral Magnitude Calculation

The data in each analysis buffer is windowed with a Hamming window. The buffer length is then doubled by extending with zeros. Padding with zeros is necessary since the autocorrelations required for the Levinson's recursion are obtained by inverse transforming the squared magnitude frequency spectrum. Therefore to prevent temporal aliasing due to the circular convolutional property of the DFT, the zero extension is necessary. Following augmentation the DFT of the buffer is taken and the spectral magnitude is computed:

$$|X(k)| = (X_R^2(k) + X_I^2(k))^{1/2} \quad k = 0, 1, \ldots, L - 1$$

where

$$X_R(k) + jX_I(k) = \mathrm{DFT}\{x(j)\}$$

Magnitude Averaging

As is shown in Chapter VI, the variance of the noise spectral estimate is reduced by averaging over as many spectral magnitude sets as possible. However the non-stationarity of the speech limits the total time interval available for local averaging. The number of averages is limited by the number of analysis windows which can be fit into the stationary speech time interval. If only reflection coefficients are to be estimated, a 128 point analysis can be used, resulting in a five set average over a 384 point stationary speech time interval. If both reflection coefficients and pitch are to be estimated from the same analysis window, then a 256 point window must be used, resulting in two set average.

Noise Bias Estimation

The SABER method requires an estimate of the expected value of the noise magnitude spectrum, μ_N:

$$\mu_N = E\{|N|\}$$

This estimate is obtained by averaging the signal magnitude spectrum $|X|$ during non-speech activity. Estimating μ_N in this manner places certain constraints when implementing the method. If the noise remains stationary during the subsequent speech activity, then an initial startup or calibration period of noise-only signal is required. During this period (on the order of a third of a second) an estimate of μ_N can be computed. If the noise environment is nonstationary then a new estimate of μ_N must be calculated prior to bias removal each time the noise spectrum changes. Since the estimate is computed using the noise-only signal during non-speech activity, a voice switch is required. When the voice switch is off an averaged noise spectrum can be recomputed. If the noise magnitude spectrum is changing faster than estimate of it can be computed, then time averaging to estimate μ_N cannot be used. Likewise if the expected value of the noise spectrum changes after an estimate of it has been computed, then noise reduction through bias removal will be less effective or even harmful.

Thus in summary, an estimate of the expected noise spectrum, μ_N, is calculated by averaging the noise signal taken during non-speech activity. This approach not only requires a speech activity, detector and a short segment of noise only signal prior to speech activity, but requires that the noise spectrum remain slowly varying with respect to the bias estimation.

Noise Bias Removal

The SABER spectral estimate $\bar{S}_A$ is obtained by subtracting the expected noise magnitude spectrum μ_N from the averaged magnitude signal spectrum $\overline{|X|}$. Thus:

$$\bar{S}_A(k) = \overline{|X(k)|} - \mu_N(k) \quad k = 0, 1, \ldots, L - 1$$

Where L=DFT buffer length.

After subtracting, the differenced values having negative magnitudes are set to some small positive value relative to the average energy. These negative differences represent frequencies where the sum of speech plus local noise is less than the expected noise. As is shown in Chapter VI, replacing negative differences with small positive values, results in an smaller spectral approximation error.

LPC Coefficient Calculation

The LPC coefficients corresponding to the spectrum $\overline{S}_A$ are calculated by initially squaring the magnitude spectrum. The inverse DFT of the resulting power spectrum is then computed. The output of this transform is a set of autocorrelations, $\hat{R}(k)$, k=0, 1, ..., L-1. A set of M LPC predictor or reflection coefficients are calculated from the first M+1 autocorrelations, using the Levinson's recursion [2]. The coefficients will represent an LPC all-pole spectral fit to the spectrum $\overline{S}_A$.

Gain Calculation

There are two choices for the gain term needed for synthesis. Either the rms of the noisy signal, g_x or the rms of the noise cancelled signal, g_s. These gain terms can be computed as:

$$g_x = (\frac{1}{L}\sum_{k=1}^{L} x^2(k))^{1/2}$$

$$g_s = (\frac{1}{L}\hat{R}(0))^{1/2}$$

where $x(k) = s(k) + n(k)$

and $\hat{R}(k) = \text{IDFT}\{\overline{S}_A{}^2\}$

It was decided that when speech activity is present that g_x is used and when speech activity is absent that g_s is used. This choice has the desirable effect of amplifying the synthesis output during speech activity and attenuating the synthesis output during non-speech activity, since g_x will be greater than g_s. The procedure for determining whether to use g_x or g_s was to examine the energy change before and after noise removal. Let R equal the amount of energy reduction in dB. Then:

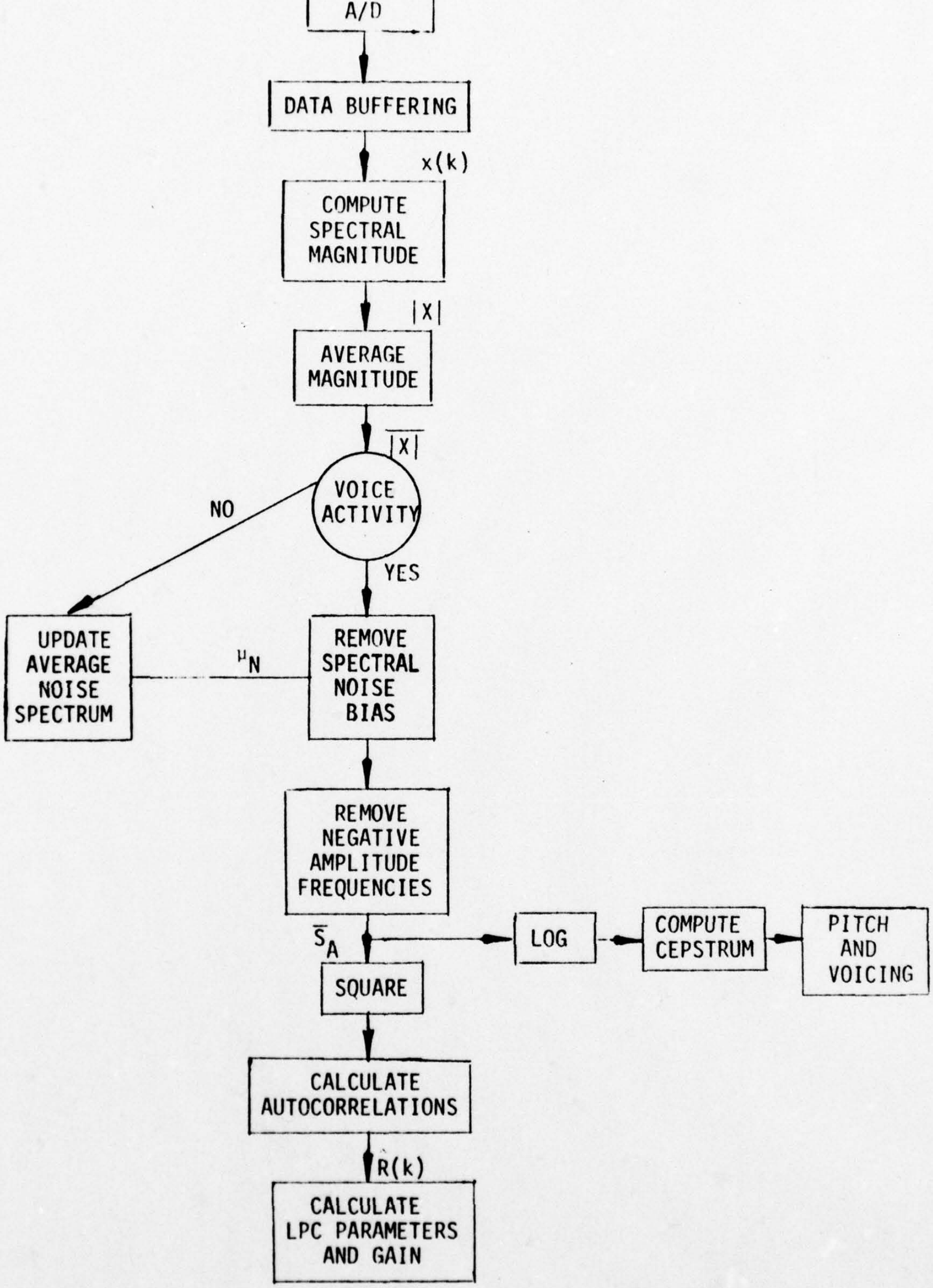
$$R = 20 \log_{10} \frac{g_x}{g_s}$$

During the non-speech noise bias estimation time period, values of R taken each analysis frame are averaged together giving an average estimate of noise power reduction $\overline{R}$. As is shown in Section VI, this average value for white, Gaussian noise is about 15dB. Speech activity is detected by comparing the current value of R with $\overline{R}$. If the current value of R is within 5dB of the average, the g_s is picked as the gain term. If the current value is smaller than the average by 5dB or more, then g_x is used. Again the reasoning behind this procedure is that in the absence of speech activity the power reduction should be largest and near its expected value. With speech present, a larger percentage of the spectrum is now speech, thus subtracting off the noise bias will only slightly reduce the total power. Thus $g_s \sim g_x$ and $R << \bar{R}$.

There are difficulties with this detection procedure however. If the signal-to-noise ratio is low $R << \bar{R}$ even during speech activity. This results in the wrong gain term being chosen and the speech synthesis is attentuated rather than amplified. Likewise at the other extreme if the noise reduction value for the current frame drops 5dB below the average during non-speech activity, the synthesis will be amplified. The 5dB value was an empirically determined threshold.

Pitch Detection

When the noise contains periodic components in the frequency range used for pitch detection, the pitch tracker can track the noise rather than the pitch. If these harmonics are first removed by the SABER method, then a pitch tracker which uses spectral magnitude information can be used to extract the actual pitch period. One such pitch detection scheme is the cepstral pitch tracker [3] After computing $\bar{S}_A$ the log is taken followed by a DFT. During voiced speech, the real cepstrum will exhibit a spike at a distance from the origin equal to the pitch period. If pitch detection is to be done based on the SABER output spectrum, then a sufficiently long analysis time window is required. A 256 point analysis window based on a 6.67 KHz sampling rate was used for cepstral pitch tracking.


A/D
DATA BUFFERING
x(k)
COMPUTE SPECTRAL MAGNITUDE
|X|
AVERAGE MAGNITUDE
$\overline{|X|}$
VOICE ACTIVITY
NO
YES
UPDATE AVERAGE NOISE SPECTRUM
μ_N
REMOVE SPECTRAL NOISE BIAS
REMOVE NEGATIVE AMPLITUDE FREQUENCIES
$\overline{S}_A$
SQUARE
LOG
COMPUTE CEPSTRUM
PITCH AND VOICING
CALCULATE AUTOCORRELATIONS
R(k)
CALCULATE LPC PARAMETERS AND GAIN

System Block Diagram

Chapter IV

Results

Introduction

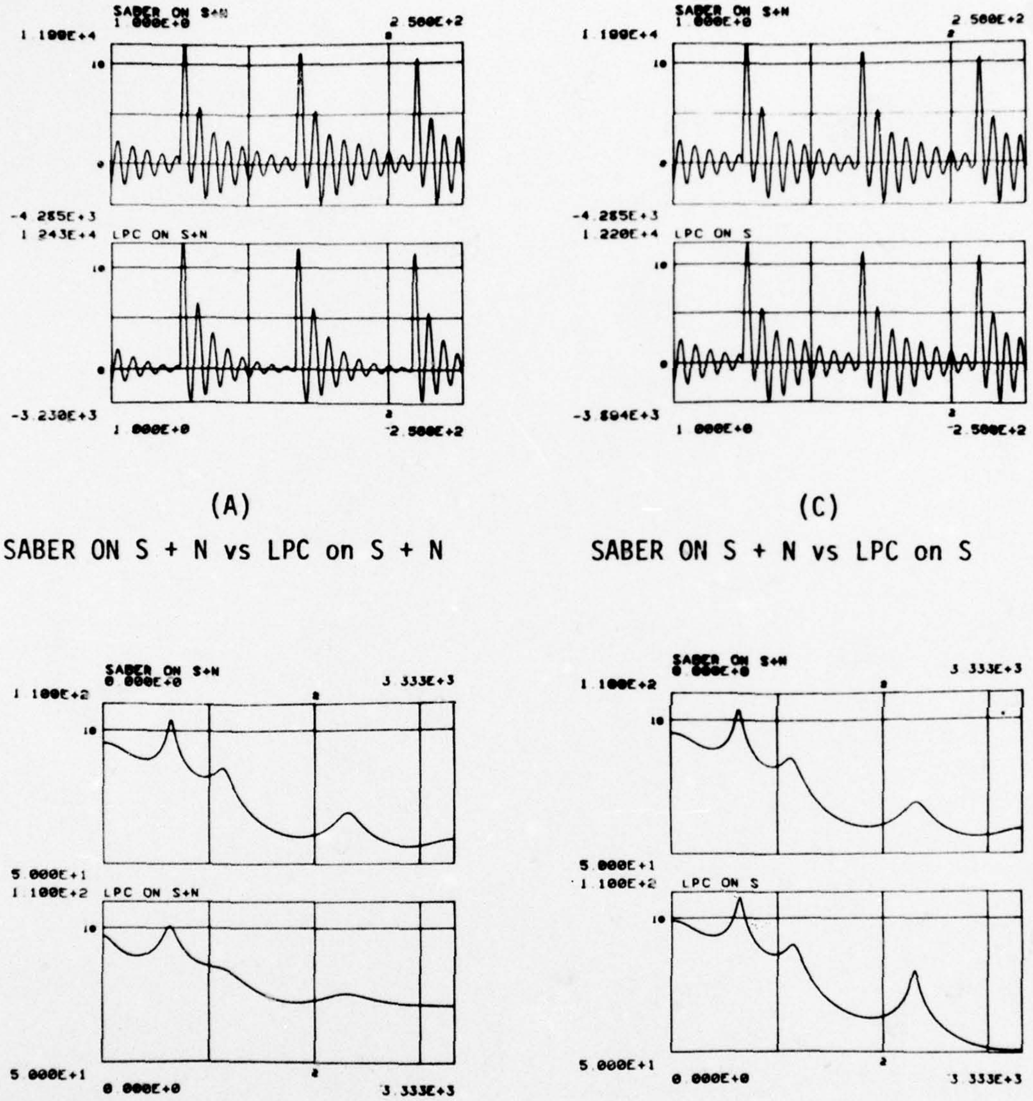
This chapter presents the results of three experiments designed to demonstrate the improvements in noise deduction and spectral resolution solving from the application of the SABER algorithm to speech analysis. Two types of noisy speech were used. In experiments one and two Gaussian noise was digitally added to clean text to produce two data bases having specified signal-to-noise ratios of 10dB and 25dB. These controlled data bases allowed for access to both the clean and noisy speech. In experiment three speech recorded in a helicopter environment with acoustically added noise was used [4]. Synthetic speech was generated from each data base. Results consist of synthesis time waveforms and corresponding all-pole spectra when standard linear prediction analysis and SABER analysis are used.

Experiment on 10dB SNR Data Base

To determine the improvement in spectral resolution obtainable from the SABER algorithm, a controlled data base was constructed. Broad band Gaussian noise was digitized from a standard analog noise generator. Clean text was recorded in an acoustically shielded sound proof room having an ambient noise level of 27dB. Both speech and noise were filtered as 3.2KHz and sampled at 6.67KHz. The average signal energy of each file was measured [4] and the noise was scaled and added to the speech to give an average signal-to-noise ratio of 10dB. The file energy calculation was taken over both speech and silent intervals. The vocoder analysis-synthesis program was modified to process either the clean or noisy speech. Three types of synthetic speech were generated: LPC on clean speech (LPC on S); LPC on the noisy speech (LPC on S+N); and SABER on the noisy speech (SABER on S+N). Note in the absence of noise, as shown in Chapter VI, SABER reduces to a standard LPC analysis, and therefore SABER on clean speech was not generated.

The first set of figures, Figure IV.1 A through D show synthetic waveforms and their all-pole spectra analyzed from the vowel in the word "dogs". Part B shows that second and third formants are clearly resolved using SABER, while considerably obscured using LPC. However, part D shows that complete noise cancellation was not achieved. This was to be expected since the clean speech had a SNR in excess of 10+15 or 25dB.

Figures IV.2 A through D show synthetic waveforms and their all-pole spectra analyzed from the fricative |sh| in "shade". Again better but not perfect spectral resolution is achieved by the SABER algorithm over standard LPC.



(A)

SABER ON S + N vs LPC on S + N

(C)

SABER ON S + N vs LPC on S

(B)

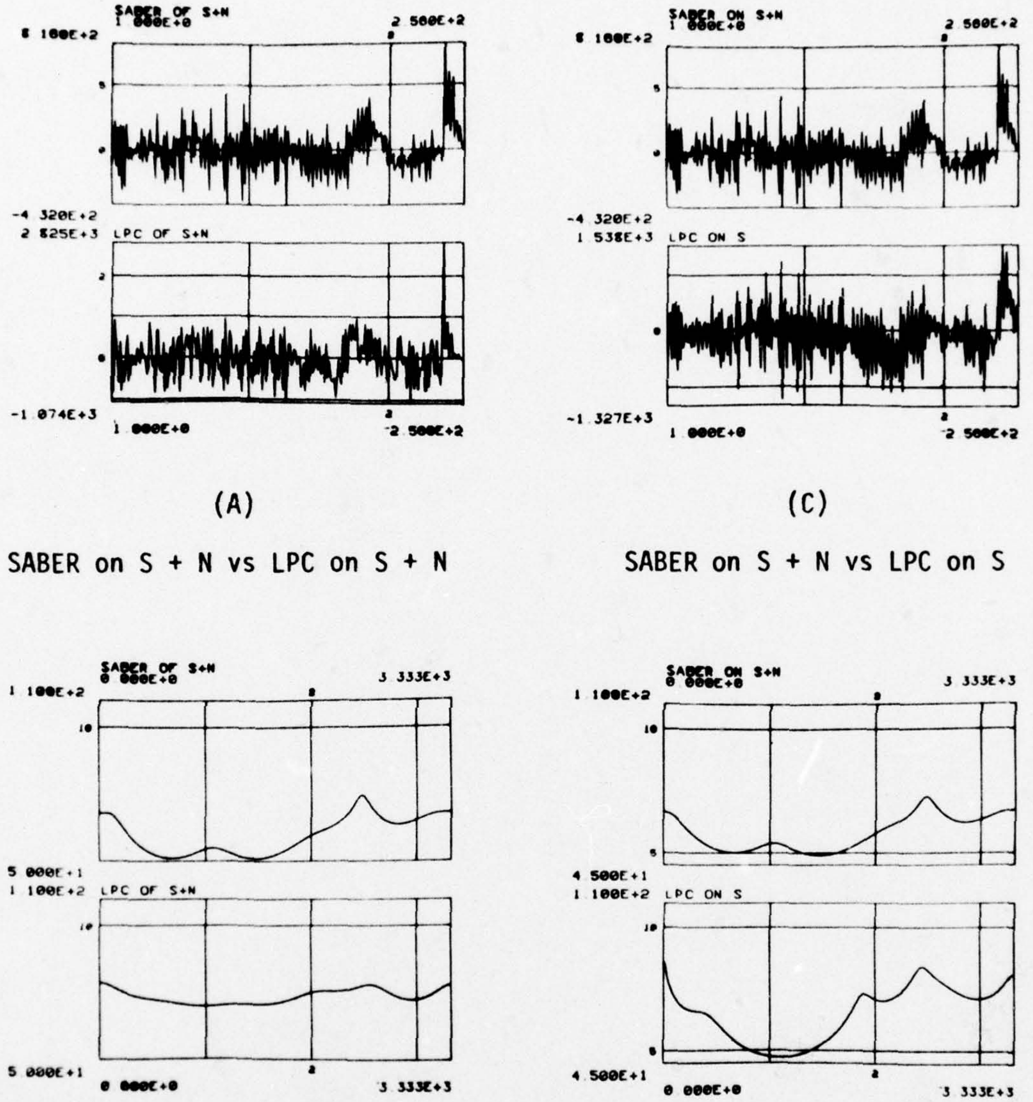
SABER ON S + N vs LPC on S + N

(D)

SABER ON S + N vs LPC on S

Voiced Time and Frequency Pairs for 10 dB SNR

Figure IV.1 A - D



(A)

SABER on S + N vs LPC on S + N

(C)

SABER on S + N vs LPC on S

(B)

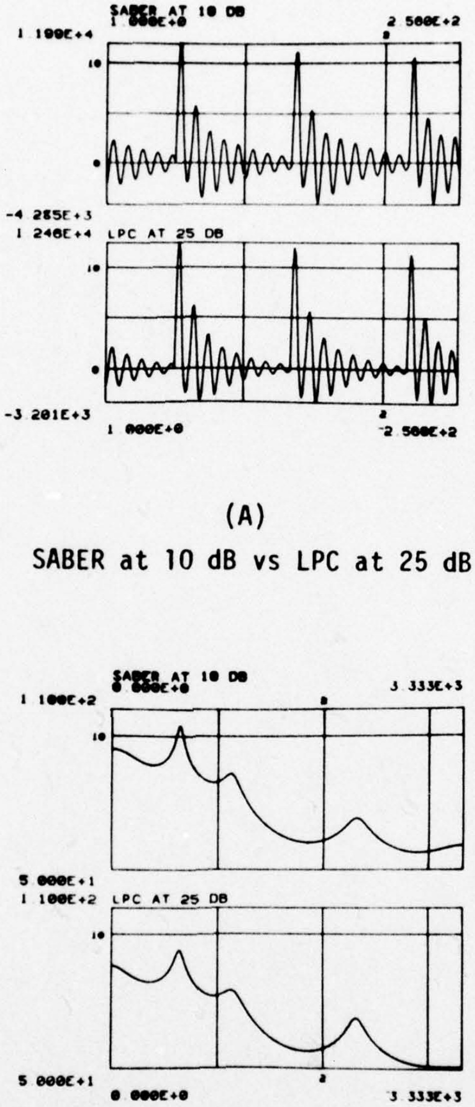
SABER on S + N vs LPC on S + N

(D)

SABER on S + N vs LPC on S

Unvoiced Time and Frequency Functions for 10 dB SNR

Figure IV.2 A - D



(A)

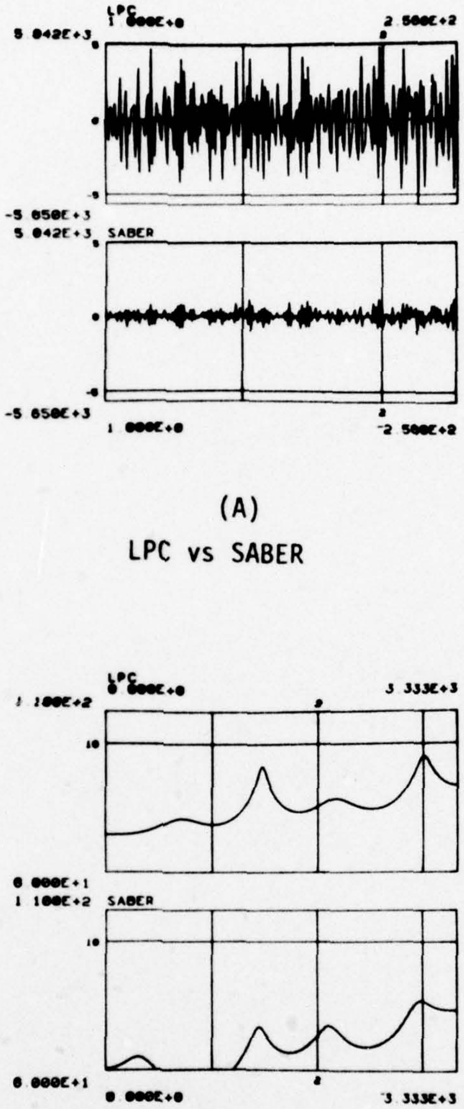
SABER at 10 dB vs LPC at 25 dB

(B)

SABER at 10 dB vs LPC at 25 dB.

Time and Frequency Functions for 10 dB vs 25 dB.

Figure IV.3



(A)

LPC vs SABER

(B)

LPC vs SABER

Time and Frequency Pairs of Helicopter Noise

Figure IV.4

Comparisons Between 10dB and 25dB SNR

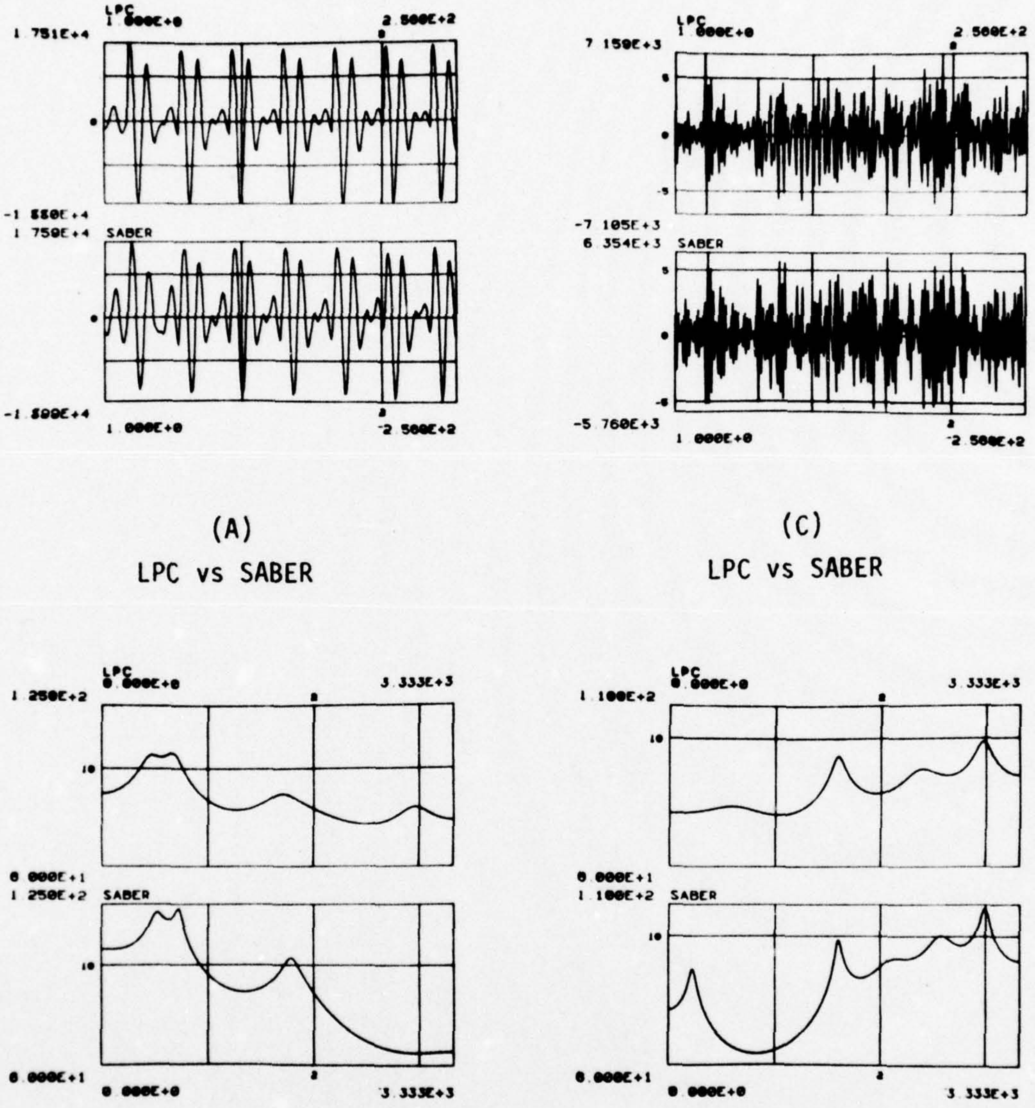
A fundamental question when evaluating the amount of noise rejection is: "After processing, how much improvement was there in the signal-to-noise ratio?" Since signal and noise energies cannot be separated and measured after processing, it is not possible to measure an SNR improvement directly. However the following indirect method can be used. If as is shown in Chapter VI, the amount of noise rejection is 15dB for white Gaussian noise, then synthetic speech generated by the SABER algorithm using the 10dB SNR data base should sound approximately equal to the synthetic speech generated by LPC using the 25dB SNR data base.

Such an experiment was conducted and the results are given in Figure IV.3 A and B. Examining Figure B shows that the all-pole spectra for LPC at 25dB has less overall energy. This is to be expected since less noise was added to the clean text to arrive at a 25dB SNR. Other than the gain difference, the spectra are approximately equal. Informal listening tests supported the results shown here, in that LPC synthesis with a 25dB input SNR was essentially indistinguishable to within a gain factor from SABER synthesis with a 10 dB input.

Experiments on Helicopter Speech

An audio test tape used in the National Security Agency's consortium testing, containing speech recorded in a helicopter environment was processed. The speech was filtered at 3.2KHz and sampled at 6.67KHz. Synthesized speech was generated using LPC and using SABER. This experiment represents true field conditions since the noise and speech are acoustically added at the microphone. Figure IV.4 A and B show time and frequency functions for LPC and SABER syntheses during noise only input. Figure A shows synthetic waveforms using the same vertical scale. The average energy difference was measured at 15.1dB. Figure B shows the all-pole spectra. The periodic harmonics at multiples of 750Hz are clearly evident in LPC spectrum (upper trace) with the second and fourth harmonics dominant at about 92dB. The all-pole spectrum corresponding to the SABER spectral estimate is given in the lower trace.

Figure IV.5 A through D show time-frequency pairs for the vowel |u| in "squirrels" and the fricative |sh| in "bushy". Note in Figure B that the SABER spectral estimate clearly separates the low first and second formants as well as shapening the third formant at about 1800 Hz. Also the noise peak at 3KHz present in the LPC spectrum is now absent in the SABER spectrum.



(A)
LPC vs SABER

(C)
LPC vs SABER

(B)
LPC vs SABER

(D)
LPC vs SABER

Voiced and Unvoiced Time Frequency Pairs from Helicopter Speech

Figure IV.5

Chapter V

Analysis of the Phase Independent Model

Introduction

Basic to the understanding of the SABER algorithm is the result that magnitude of the noisy speech spectrum can be accurately approximated by the sum of the magnitudes of speech and noise. This chapter describes this phase independent model and develops an error analysis for judging the effectiveness of the approximation. For notational convenience upper case symbols will denote Fourier transforms and lower case symbols their inverse transforms.

Thus

$$X = X(e^{j\omega}) = \sum_{k=-\infty}^{\infty} x(k)e^{-j\omega k}$$

and

$$x = x(k) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})e^{\,j\omega k}d\omega$$

Additive Noise Model and Zero Phase Approximation

Assume that a noise signal n, has been added to a speech signal s, with their sum denoted as x. Then

$$x = s + n$$

Taking the Fourier transform gives

$$X = S + N$$

The desired speech spectral magnitude, $|S|$ is given by

$$|S| = |X - N|$$

with its squared magnitude

$$|S|^2 = SS^* = |X|^2 + |N|^2 - 2|X||N|\cos(\theta_X - \theta_N)$$

where * denotes complex conjugate, θ_X the phase of X and θ_N the phase of N.

The zero phase approximation S_z to $|S|$ is given by

$$S_z = |X| - |N|$$

with its squared magnitude $|S_z|^2 = |X|^2 + |N|^2 - 2|X||N|$

Error Analysis of Zero Phase Approximation

The primary consideration of the speech analyzer is an accurate estimation of the squared magnitude function. Taking the difference D between $|S|^2$ and $|S_z|^2$ gives

$$D = |S|^2 - |S_z|^2 = 2|X||N|(1 - \cos(\theta_X - \theta_N))$$

or

$$D = - XN^* - NX^* + 2|X||N|$$

A form for D which explicitly shows its relation to S and N can be developed by noting that D can be written in the form of a perfect square.

Thus

$$D = -\left[(XN^*)^{1/2} - (NX^*)^{1/2}\right]^2$$

or

$$D = -\left[\sqrt{(S+N)N^*} - \sqrt{(S+N)^*N}\right]^2$$

After some manipulation D can be written in the following form which exnlicitly shows its dependence on the signal-to-noise ratio $|S/N|$.

$$D = -|N|^2 \left[\sqrt{|S/N| e^{j(\theta_s - \theta_N)} + 1} - \sqrt{|S/N| e^{j(\theta_N - \theta_S)} + 1} \right]^2$$

In this unfortunately complicated form, the zero phase approximation error D, can be analyzed as a function of the signal-to-noise ratio.

Extreme Error Values

Again remember that all symbols are functions of radian frequency ω. The error D will of course by zero whenever S and or N are zero.

A worse case condition, that is when D is maximum, will occur when

$$\theta_N - \theta_S = \pm \pi/2$$

Then

$$D = - |N|^2 \left[\sqrt{1 - j|S/N|} - \sqrt{1 + j|S/N|} \right]^2$$

or after squaring

$$D = 2|N|^2\left[\sqrt{|S/N|^2 + 1} - 1\right]$$

For

$$|S/N| << 1 \quad D \approx 0$$

For

$$|S/N| >> 1 \quad D \approx 2|N||S|$$

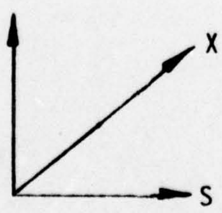
The relative error is

$$D/|S|^2 = 2|N|/|S| << 1$$

Finally, for $|S| \approx |N|$

$$D \approx 2|N|^2\left[\sqrt{2} - 1\right]$$

This situation is depicted graphically as



Here the relative error is given by

$$D/|S|^2 \approx 2\left[\sqrt{2} - 1\right] \approx -6\text{dB}.$$

Summary

This chapter investigated the spectral error induced by using a zero phase approximation to the magnitude spectrum. The worse case condition occurs when the magnitudes of S and N are equal and out of phase by ninety degrees. For other situations the error was negligible or small compared to the speech spectrum. Informal listening tests judged the LPC synthesis based upon the zero phase power spectrum to be essentially indistinguishable from LPC synthesis based upon the actual power spectrum. Although differences can be detected, they become inconsequential compared to the noise cancellation capabilities provided by using the zero phase approximation. The application of the zero phase model to noise suppression is developed in the next chapter.

Chapter VI

Analysis and Reduction of Estimation Error using the SABER Method

Introduction

Using the result of Section V that the magnitude of speech plus noise can be accurately approximated by the sum of the zero phase estimate of speech and noise magnitude, the following linear model is implied:

$$|X| = S_Z + |N|$$

where

S_Z = zero phase approximation to the magnitude of the Fourier transform of the windowed speech, s(k)

$|N|$ = Magnitude of the Fourier Transform of the Additive windowed noise n(k).

and

$|X|$ = Magnitude of the spectrum of s(k) + n(k)

Again upper case symbols denote Fourier transforms of lower case symbols. This model shows that the noise spectrum, $|N|$ enters in as a spectral bias added to the desired speech spectrum, S_Z. The more accurate this bias can be estimated, the more accurate will be the resulting estimated spectrum obtained by subtracting the bias estimator from the noisy speech magnitude, $|X|$. This section describes an approach to bias estimation and removal using short time averaging.

The Non-Averaged SABER Estimate

Assume that the additive noise n(k) is stationary and thus that the expected value of the noise magnitude:

$$E\{|N|\} = \mu_N$$

is constant over time. (In practice μ_N is estimated by a time average taken during non-speech activity.)

A spectral estimate S_A of S_Z can be defined as

$$S_A = |X| - \mu_N$$

The error ε in approximating S_Z by S_A is

$$\varepsilon = S_A - S_Z = |X| - \mu_N - |X| + |N|$$

or

$$\varepsilon = |N| - \mu_N$$

Using this estimate the value of the error equals the difference between the magnitude of the noise spectrum and its expected value.

Reduction in Error Through Averaging: The SABER Spectral Estimate

A straight forward method for reducing the spectral error ε is through averaging. Averaging magnitude spectra $|X(i)|$ taken from possibly overlapping time windows gives

$$\overline{|X|} = \frac{1}{k} \sum_{i=1}^{k} |X(i)| = \frac{1}{k} \sum_{i=1}^{k} S_Z(i) + |N(i)|$$

or

$$\overline{|X|} = \bar{S}_Z + \overline{|N|}$$

The SABER spectral estimate is given by:

$$\bar{S}_A = \overline{|X|} - \mu_N$$

The spectral error $\bar{\varepsilon}$ in approximating $\bar{S}_Z$ by $\bar{S}_A$ is

$$\bar{\varepsilon} = \bar{S}_A - \bar{S}_Z = \overline{|X|} - \mu_N - \overline{|X|} + \overline{|N|}$$

or

$$\bar{\varepsilon} = \overline{|N|} - \mu_N$$

Assuming first that the zero phase approximation S_Z is an accurate approximation of $|S|$ as argued in Chapter V and second that during the total time segment over which the averages are taken that the speech spectra $S_Z(i)$ remain essentially constant, then

$$|S| \approx S_Z \approx \bar{S}_Z$$

Thus the averaged spectral magnitude error $\bar{\varepsilon}$ represents

$$\bar{\varepsilon} = \overline{|N|} - \mu_N \approx \bar{S}_A - |S|$$

Assuming stationarity this shows that as more time-averaged spectra are used, the sample mean $\overline{|N|}$ will converge to μ_N and $\bar{S}_A$ will converge to $|S|$. Unfortunately the nonstationarity of the speech limits the time interval over which averages can be taken. Thus the error can only be reduced to the extent to which the sample mean $\overline{|N|}$ has converged to the mean μ_N.

In terms of mean squared error, the variance of the error

$$\sigma_\varepsilon^2 = \mathrm{var}(\bar{\varepsilon}) = E\{(\bar{S}_A - \bar{S}_Z)^2\}$$

equals the variance of the sample mean of the magnitude noise spectrum:

$$\sigma^2_{\overline{|N|}} = \mathrm{VAR}(\overline{|N|}) = E\{(\overline{|N|} - \mu_N)^2\}$$

The error will be minimized to the same extent that the variance of the sample mean has been minimized.

In summary, the noise-suppression, signal estimation problem, using the zero phase approximation linear model, can be reduced to a spectral estimation problem. Unfortunately due to the nonstationarity of the speech, only a limited number of spectral time averages can be used to minimize the variance of the sample mean. Thus complete noise cancellation is not possible. At this point justification for the name SABER is apparent. The noise spectrum shows up as a bias on the desired signal. To remove the bias, averages are taken for as long as the underlying speech remains stationary. Averaging will reduce the variance of the local noise bias. The bias can be partially removed by subtracting off the expected value of noise bias. The smaller the variance, the better the noise reduction.

Expected Value of Noise Reduction

When the additive noise n(k) is zero mean, white, and Gaussian, an estimate of the expected value of noise reduction can be computed. In turn the amount of noise reduction can be equated to the variance of the error between the SABER estimate, and the speech spectrum. Specifically, from the last chapter let

$$\bar{\epsilon} = \bar{S}_A - \bar{S}_Z = \overline{|N|} - \mu_N$$

Define σ_n^2 as the variance or average power of the additive noise. Then

$$\sigma_n^2 = E\{n^2(k)\}$$

Using the results of Appendix B, the expected value of the noise magnitude, μ_N, having a χ distribution of order two will equal:

$$E\{|N|\} = \mu_N = \sigma_n \sqrt{\frac{\pi L}{4}} \qquad L = \text{analysis window length}$$

The squared noise spectral magnitude, having a χ^2 distribution of order 2 will equal $\sigma_n^2 L$:

$$E\{|N|^2\} = \sigma_n^2 L$$

Therefore the variance of the magnitude will equal

$$E\{(|N| - \mu_N)^2\} = E\{|N|^2\} - \mu_N^2 = \sigma_n^2(1 - \frac{\pi}{4})L$$

Without averaging, by simply subtracting the mean from the magnitude spectrum, the expected value of noise reduction will equal 6.68dB:

$$10 \log_{10} \frac{E\{|N|^2\}}{E\{(|N| - \mu_N)^2\}} = 6.68 \text{ dB}$$

Further Reduction in Variance Through Averaging

The noise variance can be reduced by averaging magnitude spectra taken from possibly overlapping time windows. This technique has been carefully investigated by Welch [5]. A summary of his variance analysis is described in Appendix A. Again because of the nonstationarity of the speech spectra, only a limited time interval is available for averaging. Using Welch's formulation, the variance is minimized by first specifying the type and length of window required, second the time interval available for averaging, and then averaging magnitude spectra based upon windowed data overlapped by one-half a window length. For this implementation, a 19.2 ms Hamming window was used with a time interval of 57.6 ms. This allowed for a total of five windows to be used. The variance of the sample mean after averaging equaled

$$E\{(\overline{|N|} - \mu_N)^2\} = 0.275\ \mathrm{Var}\{|N|\} = (0.275)(.21)\sigma_n^2 L$$

The expected value of noise reduction is

$$10 \log_{10} \frac{\sigma_n^2 L}{(0.275)(0.21)\sigma_n^2 L} = 12.4 \text{ dB}$$

Discussion

The choice of window length and averaging interval values represent a compromise in conflicting requirements. For acceptable spectral resolution a window length greater than twice the expected largest pitch period is required [1]. For minimum variance a large number of windows are required for averaging. Finally, for acceptable time resolution a narrow analysis interval is required. Using a Hamming window the effective averaging time interval reduces from 57.6 ms to approximately 39 ms due to the window edge attenuation. A 19.2 ms window length has been found to result in acceptable frequency resolution [6]. Thus to achieve better noise reduction some compromise in both time and frequency resolution was necessary.

Negative Magnitude Stripping

An additional technique for reducing the spectral error is to replace $\bar{S}_A$ with zero when ever the difference goes negative. During non-speech activity this will on the average reduce the resulting noise power in half for an additional 3dB improvement. When speech is present and μ_N is larger then $\overline{|X|}$ the spectral error between S_Z and $\bar{S}_A$ will be larger than if zero is used for the estimate. That is for:

$$\overline{|X|} < \mu_N$$

have

$$-\bar{S}_A > 0$$

and

$$(S_Z - \bar{S}_A)^2 > (S_Z - 0)^2$$

Summary

This section developed an equality between the error in the SABER spectral estimate and the variance of the sample mean of the additive noise spectral magnitude. For additive noise which is white, zero mean, and Gaussian, the expected value of noise reduction resulting from averaging, subtracting off the mean and zeroing out negative difference components was approximately 15dB.

Appendix A

Variance Reduction Through Averaging Magnitude Spectra

Introduction

It was shown in Chapters V and VI that the noise spectrum shows up as a bias on the desired signal. To remove the bias, averages are taken for as long as the underlying speech remains stationary. Averaging will reduce the variance of the local noise bias. The bias can be partially removed by subtracting off the expected value of noise bias. The smaller the variance, the better the noise reduction. Due to the nonstationarity of the speech only a limited time interval is available for averaging. This appendix reviews an analysis published by Welch [5] for determining the variance reduction possible as a function of averaging interval, M, window length, L, window shape $w(j)$, $j = 0, 1, \ldots, L - 1$, and overlap interval D.

Using the results of this analysis the amount of variance reduction was calculated.

Data Segmentation

Define $x(j)$ $j = 0, 1, \ldots, M - 1$ to be samples of a second order, stationary stochastic sequence. Data segments $x_i(j)$, possibly overlapping of length L are defined with starting points D units apart. Thus define

$$\begin{aligned} x_1(j) &= x(j) \\ x_2(j) &= x(j + D) \qquad j = 0, 1, \ldots, L - 1 \\ &\vdots \\ x_K(j) &= x(j + (K - 1)D) \end{aligned}$$

Assume there are K segments covering the entire averaging interval:

$$(K - 1)D + L = M$$

Magnitude Spectrum Calculation and Averaging

Using a data window, w(j) (for this analysis a Hamming window was selected), the windowed data sequences are formed:

$$\{x(j)w(j)\}, \ldots \{x_k(j)w(j)\}$$

For each windowed data sequence, the discrete Fourier transform is calculated:

$$X_k(\ell) = \sum_{i=0}^{L-1} x_k(i)w(i)e^{-j \frac{2\pi}{L} i\ell} \qquad \ell = 0, 1, \ldots, L - 1$$

followed by the magnitude spectrum:

$$|X_k(\ell)| = (\mathrm{Re}^2\{X_k(\ell)\} + \mathrm{Im}^2\{X_k(\ell)\})^{1/2}$$

Averaging the spectral estimates gives

$$\overline{|X(\ell)|} = \frac{1}{K}\sum_{k=1}^{K} |X_k(\ell)|$$

Variance of $\overline{|X|}$

Define the covariance of $\overline{|X|}$ as

$$d(j) = \mathrm{cov}\{|X_k(\ell)|, |X_{k+j}(\ell)|\}$$

Then it can be shown that

$$\mathrm{var}\{\overline{|X(\ell)|}\} = \frac{1}{K}\left\{d(0) + 2\sum_{j=1}^{K-1} \frac{K-j}{K}\, d(j)\right\}$$

Further, defining the correlation of $\overline{|X|}$ as

$$\rho(j) = \text{correlation}\{|X_k(\ell)|, |X_{k+j}(\ell)|\} = \frac{d(j)}{d(0)}$$

then

$$\text{var}\{\overline{|X(\ell)|}\} = \frac{\text{var}\{X_k(\ell)\}}{K} \left\{1 + 2\sum_{j=1}^{K-1} \frac{K-j}{K}\,\rho(j)\right\}$$

Assuming that the spectrum of x(j) is flat and Gaussian, the correlation $\rho(j)$ of $\overline{|X|}$ is given by

$$\rho(j) = \sum_{k=0}^{L-1} w(k)w(k + jD) / \sum_{k=0}^{L-1} w^2(k)$$

Minimum Variance Determination

For a fixed averaging interval, M, window length, L, and window shape w(j), the optimal spacing D can be calculated to minimize the expression:

$$F = \frac{1}{K} \left\{ 1 + 2 \sum_{j=1}^{K-1} \frac{K-j}{K} \rho(j) \right\}$$

Note that $\rho(j)$ is greater than or equal to zero. For $D \geq L$ (no overlap) $\rho(j) = 0$ and the variance will reduce as 1/K. By overlapping segments, K increases at the expense of an increasing $\rho(j)$. After some calculation the minimum as suggested by Welch, was found to occur for D = L/2, that is, overlap by one-half a window length.

When using the following parameters imposed by the speech analysis constraints:

$M = 384$

$L = 128$

$D = 64$

$K = 5$

the resulting value for F was calculated to be 0.275.

Appendix B

Calculation of the Mean and Variance

Introduction

This appendix develops estimates for the mean and variance of the magnitude spectrum for white, zero-mean Gaussian noise. Of course, this development is not original, with this version taken in part from Ingebretsen [7].

Mean and Variance of the Fourier Transform

Let x(k) be a zero mean, white Gaussian sequence with $X(e^{j\omega})$ its Fourier transform. Then

$$X(e^{j\omega}) = X_R(e^{j\omega}) + jX_I(e^{j\omega}) = \sum_k x(k)e^{-jk\omega}$$

where X_R and X_I are the real and imaginary parts of X. $X_R$and X_I will be normal since x(k) is normal. In addition

$$E\{X_R\} = E\{X_I\} = 0$$

It can be shown [8] that

$$\mathrm{cov}\{X_R(e^{j\omega_1}),\ X_R(e^{j\omega_2})\} = \mathrm{cov}\{X_I(e^{j\omega_1}),\ X_I(e^{j\omega_2})\} = 0$$

for $\omega_1 \neq \omega_2$

Thus the vector pairs $(X_R(e^{j\omega_1}),\ X_R(e^{j\omega_2}))$ and $(X_I(e^{j\omega_1}),\ X_I(e^{j\omega_2}))$ are independent since X is Gaussian. Likewise, X_R and X_I are independent of each other.

Using the independence of x(k), the variance of the real and imaginary part of $X(e^{j\omega})$ is given by

$$\mathrm{var}\{X_R(e^{j\omega})\} = E\{\sum_k x^2(k)\cos^2(k\omega)\}$$

or

$$\mathrm{var}\{X_R(e^{j\omega})\} = \sigma^2 \sum_k \cos^2(k\omega)$$

where

$$E\{x^2(k)\} = \sigma^2$$

Assuming the transform is taken using a window of length L samples then

$$\sum_{k} \cos^2(k\omega) = \frac{L}{2} + \frac{1}{2}\,\frac{\text{SIN } L\omega}{\text{SIN } \omega}$$

Evaluating the transform at equally spaced frequencies $\frac{2\Pi}{L}$ results in:

$$\text{var}\{X_R(e^{j\omega})\} = \frac{L\sigma^2}{2} \quad \omega = \frac{k2\pi}{L} \quad k = 0, 1, \ldots, L - 1.$$

Likewise

$$\text{var}\{X_I(e^{j\omega})\} = \frac{L\sigma^2}{2} \qquad \omega = \frac{k2\pi}{L} \quad k = 0, 1, \ldots, L - 1.$$

Expected Value of Magnitude and Squared Magnitude of the Fourier Transform

The frequency magnitude is equal to the square root of the sum of squares of two independent, normal random variables. It thus has a χ distribution:

$$|X| = (X_R^2 + X_I^2)^{1/2} \qquad \omega = \frac{k2\pi}{L} \qquad k = 0, 1, \ldots, L - 1$$

The expected value of $\overline{|X|}$ is found by evaluating the integral:

$$E\{|X|\} = \frac{1}{\frac{L\sigma^2}{2}} \int_0^\infty x^2 e^{-x^2/L\sigma^2} dx$$

Evaluating gives

$$E\{|X(e^{j\omega})|\} = \sigma\sqrt{\frac{L\pi}{4}} \qquad \omega = \frac{k2\pi}{L} \qquad k = 0, 1, \ldots, L - 1$$

The squared magnitude will have a χ^2 distribution. Its expected value is given by

$$E\{|X|^2\} = \frac{1}{L\sigma^2}\int_0^\infty ye^{-y/L\sigma^2}dy$$

Evaluating gives:

$$E\{|X(e^{j\omega})|^2\} = L\sigma^2 \qquad \omega = \frac{k2\pi}{L} \qquad k = 0, 1, \ldots, L - 1$$

This last expression can also be obtained from Parseval's relation for the Discrete Fourier Transform:

$$\sum_{k=0}^{L-1} x^2(k) = \frac{1}{L}\sum_{\ell=0}^{L-1} |X(\ell)|^2$$

REFERENCES

[1] Makhoul, J., Wolf, J. "Linear Prediction and the Spectral Analysis of Speech," NTIS No. AD-749066, BBN Report No. 2304, Bolt Beranek and Newman, Inc., Cambridge, Mass. 1972.

[2] Markel, J., Gray, A. "Linear Prediction of Speech", New York, Springer Vellag, 1976.

[3] Noll, A.M., "Cepstrum Pitch Determination", J. Acoust. Soc. Amer., Vol. 41, pp 293-309, Feb. 1967.

[4] Boll, S.F., "Noise Suppression Methods for Robust Speech Processing", Semi-Annual Technical Report, UTEC-CSc-77-090, Computer Science Dept., University of Utah, April 1977.

[5] Welch, P., "The Use of the Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodograms", IEEE Trans. Audio Electoacoust. Vol. AU-15, pp. 70-73, June 1967.

[6] Cohen, D., "Specifications for the Network Voice Protocol", ISI/RR-75-39, Information Sciences Institute, Univ. of Southern California, March 1976.

[7] Ingebretsen, R., "Log Spectral Estimation for Stationary and Nonstationary Processes", UTEC-CSc-75-118, Computer Science Dept., University of Utah, Aug. 1976.

[8] Jenkins, G.M., Watts, D.G., "Spectral Analysis and Its Applications", San Francisco, Holden-Day, 1968.

Section II

Current Results on Dual Input Nonstationary Noise Suppression Using LMS Adaptive Noise Cancellation

Dennis Pulsipher

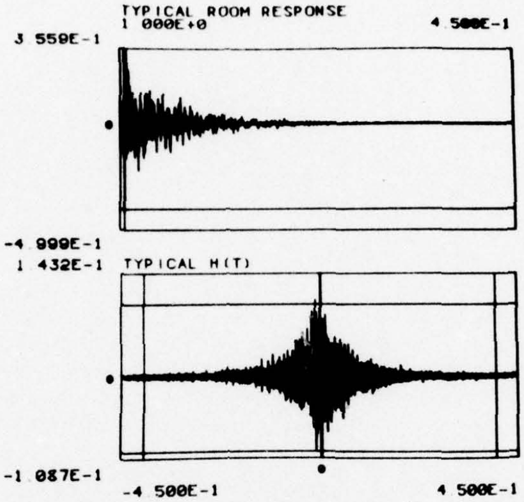
INTRODUCTION

I. The preceding semi-annual technical report described a dual input noise suppression technique for audio signals. It also described several experiments performed successfully using synthetic data, data which was forced to comply with all of the underlying assumptions. The results of these experiments were quite encouraging.

Since that time the research has continued to apply this technique to actual acoustically recorded noisy speech. A brief description of recent results and conclusions comprise this section.

OBSERVATIONS

As work with acoustically recorded data with high level broad-band noise has progressed it has become increasingly apparent that a factor of major importance is the length of the required filter. A look at the impulse responses of typical hard-walled rooms indicates that the duration of significant energy frequently lasts half a second or more. (Figure 1a). Since in practice the source of the noise n is separated from both the location of the reference noise pick-up and the noisy speech signal pick-up, it is apparent that the situation is not completely described by a single room impulse response.



TOP: Typical Impulse Response of a Room (G(T))

BOTTOM: Typical Noise Cancelling Filter for the Same Room (H(T))

Comparison of a Typical G(T) and H(T)

Let G_1 be the response of the room from the noise source to the noisy speech signal pick-up position and G_2 be the room's response from the noise source to the reference noise pick up position. Then the filter to be estimated is neither G_1 nor G_2 but G_2 G_1. (Figure 2). This filter may be as long as the sum of the lengths of G_1 and G_2(less one point). (Figure 1b)

This affects significantly the estimation of the noise samples u_j .

The optimal noise estimate u_j^* is

$$u_j^* = \sum_{i=\infty}^{\infty} h^*(j-i)\nu(i)$$

In actual practice, though, the filter estimated is

$$\hat{u}_j = \sum_{i=n}^{m} \hat{h}(j-i)\nu(i)$$

The error in this estimate is due to two factors, the first is the difference caused by a finite length filter. This error can be reduced by making the filter longer. The second factor is the error in filter coefficient estimation. By increasing the number of filter taps, the variance on our noise estimate may in fact increase, and thus degrade our system performance. That is, if we increase the number of filter coefficients we must also increase the accuracy of estimation of the coefficients or the improvement in

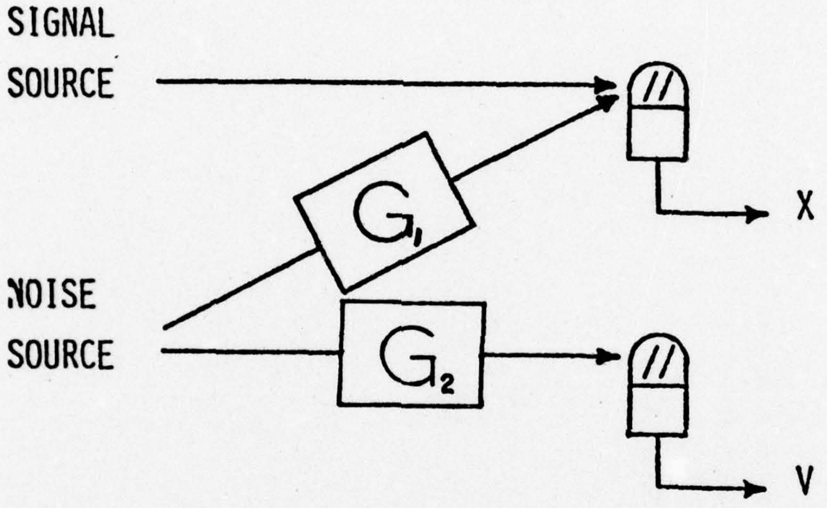


Figure 2

Data generation model

performance caused by lengthening the filter may be outweighed by the degradation caused by miss-estimation of coefficients.

This increased accuracy required by longer filter lengths may be achieved by decreasing the rate of adaptation.

Since estimating long filters reouires more computation and since a slow adaptation rate requires processing more data, it has been found that performing such experiments is a time consuming process using the non-real time simulation. Present results indicate, that for broadband noise, a minimum of 10dB noise reduction using a .4 sec. adaption filter is possible. For highly correlated noise much shorter filters (on the order of 100 taps) can be successfully used giving considerably better noise rejection.

Additional experiments with longer filters and slow adaptation rates are currently being performed. Effort is also being put into determining the optimal filter length and adaptation rates for different types of noise and different channels, and in the actual helicopter operating environment.

SECTION III

ESTIMATION OF THE PARAMETERS OF AN AUTOREGRESSIVE-MOVING AVERAGE PROCESS IN THE PRESENCE OF NOISE

William J. Done

INTRODUCTION

Two important aspects of vocoder development have been the achievement of high quality synthetic speech and the development of low bit rate communications systems. Linear prediction (LP) vocoders have gained importance in both of these areas. However, evaluations of the quality and intelligibility of LP and other vocoders are usually performed with high quality speech inputs undegraded by background noise. When noise is added to the speech signal prior to analysis, the intelligibility and quality of the synthetic speech derived from the vocoders are degraded, the results often being unacceptable. In LP vocoders, the addition of noise causes problems in four areas: 1) silence detection, 2) voiced/unvoiced determination, 3) pitch period calculation if voiced, and 4) spectral matching errors. McAulay [10] has addressed problems 1), 2), and 3). Spectral matching errors, which result from the inaccurate identification of the linear prediction parameters (autoregressive parameters) due to the effects of noise, will be the primary emphasis of this research. Results illustrating the spectral degradation due to additive white noise will be presented.

Given a time series that can be successfully modeled as a parametric process, such as an autoregressive-moving average (ARMA) process, there are primarily two approaches that can be taken to alleviate the distortion due to spectral matching errors caused by noisy data. The most often used technique is to suppress the noise prior to analysis, the analysis methods depending upon the parametric model in use. This prefiltering technique is thus seen as a two step process: noise removal followed by parameter analysis. Usually the parameter analysis step remains unchanged from the noiseless case. Examples of prefiltering include the various applications of Wiener filtering, adaptive noise cancelling techniques, or linear filtering techniques used to eliminate frequency bands dominated by noise.

The second approach to parameter estimation in the presence of noise involves the construction of a new model which explicitly accounts for the effects of the noise. Rather than the two stage technique of noise suppression--parameter extraction, the modeling approach requires a one step system in which the analysis methods for parameter extraction are significantly changed from the analysis methods used when no noise is present. The modifications are required to account for the changes in the parametric model as a consequence of adding the noise. In the case of an autoregressive (AR) process corrupted by additive Gaussian white noise, the modeling approach for parameter estimation is especially appealing.

Both approaches have advantages and disadvantages. If a model is successful with high quality signal inputs and algorithms for parametric estimation are well established, then prefiltering may be desirable when noise is present. It offers the advantage of retaining the original parameter estimation algorithm with little or no alteration. Prefiltering, however, may have a side effect of distorting the characteristics of the desired signal in the process of suppressing the noise. An example of this could be low pass filtering of the noisy data in a situation where the noise becomes dominant above some frequency f_L. Although the filtered signal may have an improved signal-to-noise ratio (SNR), some information contained in the signal is lost. Another possible disadvantage of the prefiltering approach arises when the system is to be used in an environment where the noise characteristics are changing. Here decisions must be made about the structure of the prefilter: should several types of filters, each matched to a type of noise, be used or should an adaptive filter structure be used.

If the time series to be characterized is successfully modeled by a parametric representation like the AR model, then the modeling approach to parameter extraction may be desirable if the effects of the noise can be included in a modified model. The primary disadvantage of the modeling approach is the need to change the parameter estimation procedure, requiring the development of new algorithms. For the signal-in-noise model discussed here, the addition of noise to an AR process results in an ARMA process. Because of the nonlinear relationsips involved, the task of identifying the parameters of an ARMA process is much more difficult than identifying the parameters of an AR process. In addition, there is the validity of the parametric model for the time series being considered, which may pose a limitation on the modeling technique. This approach also shares the disadvantage that occurs in nonstationary noise situations.

This section will describe a technique in which an autoregressive process of order q, AR(q), with AR parameters $\{a(i)\}_1^q$ is identified when corrupted by additive Gaussian white noise. It will be shown that the additive noise changes the time series from an AR process to an ARMA process, from which the q original AR parameters can be identified. Preliminary results are given on the initial efforts to implement the algorithms necessary for determination of the desired AR coefficients.

As stated above, one limitation of the modeling approach is the validity of the parametric model, an AR process in this case. Voiced speech has been successfully "modeled" by an all-pole LPC process. However, if voiced speech is assumed to be an AR process and identification of the AR parameters is desired, the periodic or seasonal component can hinder the "identification" of the AR parameters. The parametric model must be modified to account for the presence of the periodic component, introducing the concept of autoregressive- integrated moving average (ARIMA) processes. Extensions of the parametric model for the signal to include ARI, ARMA, and ARIMA forms will be considered. The ARI model for voiced speech waveforms will be emphasized. This model will be studied in both noiseless and noisy conditions.

SYSTEM DESCRIPTION

In this section, the procedures for parameter estimation in the presence of noise will be discussed. The effects of additive noise on LPC parameters will be described and a detailed presentation of the mathematics defining the ARMA model will be given. Of the four parts to follow, part one will present the algorithm for linear predictive coding. The effects of additive noise upon the LPC parameters can then be observed. The popular LPC approach can also be compared with the ARMA model technique.

Part two of the system description will present the ARMA model algorithm suggested by Pagano [11]. The mathematical details of this technique are given. In developing the ARMA model approach, the initial step is a stage for estimating the ARMA parameters. Once these estimates are available, a nonlinear regression modifies the initial estimates. Part three will discuss possible ARMA estimation procedures and the nonlinear regression technique. The last part will include a discussion of important software not included directly in the operation of the LPC or ARMA model analysis systems. Also discussed in this last part is the data base used for analysis.

Linear Predictive Coding

If s(k) is a time series which can be modeled as a q^{th}-order autoregressive process, AR(q), then

$$s(k) = -\sum_{i=1}^{q} a_1(i)s(k-i) + \varepsilon(k), \qquad 1)$$

where $\{a_1(i)\}_1^q$ are the AR parameters and $\varepsilon(k)$ is a zero mean white noise process. In developing the expressions characterizing LPC and presenting only the autocorrelation method of analysis, the equations are much more compact if matrix notation is used. Refer to Makhoul [9] for additional background and a list of references for LPC development. The development of a notational convention for LPC using a matrix formulation can be found in Boll [2].

Using the autocorrelation method, the sequence s(k) has infinite extent but is nonzero only for $0 \leq k \leq N-1$, where N is the size of the analysis window. Form the (N+q) x 1 vector $\underline{s}$, where

$$\underline{s} = [\ s(0)\ s(1)\ \ldots\ s(N-1)\ 0\ \ldots\ 0\ ]^T. \qquad 2)$$

Using D as a delay operator for vector notation, $D^i\underline{s}$ is an (N+q) x 1 vector with the sequence s(0) ... s(N-1) beginning at the (i+1)th position. The superscript i can take on the values 1, ..., q. For example,

$D^1\underline{s} = [0\ s(0)\ s(1)\ \ldots\ s(N-1)\ 0\ \ldots\ 0]^T$,
$D^2\underline{s} = [0\ 0\ s(0)\ s(1)\ \ldots\ s(N-1)\ 0\ \ldots\ 0]^T$, and
$D^q\underline{s} = [0\ 0\ \ldots\ 0\ s(0)\ s(1)\ \ldots\ s(N-1)]^T$.

Form the (N+q) x q matrix $\underline{H}_s$ by appending as columns the $D^i\underline{s}$, i = 1, ..., q

$$\underline{H}_s = [D^1\underline{s}\ \ D^2\underline{s}\ \ D^3\underline{s}\ \ \ldots D^q\underline{s}\ ]. \qquad 3)$$

If an error sequence $\underline{\varepsilon}$ is defined as

$$\underline{\varepsilon} = [\ \varepsilon(0)\ \ \varepsilon(1)\ \ \ldots\varepsilon(N-1)\ \ \varepsilon(N)\ \ \ldots\varepsilon(N+q-1)\ ]^T, \qquad 4)$$

then 1) can be written as

$$\underline{s} = -\underline{H}_s\underline{a}_1 + \underline{\varepsilon}, \qquad 5)$$

where $\underline{a}_1 = [\ a_1(1)\ \ a_1(2)\ \ \ldots\ a_1(q)\ ]^T$ is formed from the prediction coefficients and the index k in 1) is confined to the interval $0 \leq k \leq N+q-1$. In LPC the measure of closeness of fit is the least squares minimization of the energy in the error signal $\underline{\varepsilon}$, as a function of the $\{a_1(i)\}_1^q$. If the loss function L_ε is defined as

$$L_{\varepsilon} = \sum_{k=0}^{N+q-1} \varepsilon^{2}(k) = \underline{\varepsilon}^{T}\underline{\varepsilon}, \qquad 6)$$

then the minimum of L_{ε} with respect to the $\{a_{1}(i)\}_{1}^{q}$ is found. Using vector calculus,

$$\frac{\partial L_{\varepsilon}}{\partial \underline{a}_{1}} = \frac{\partial}{\partial \underline{a}_{1}} [\underline{\varepsilon}^{T}\underline{\varepsilon}] = 2\underline{\varepsilon}^{T} \frac{\partial \underline{\varepsilon}}{\partial \underline{a}_{1}} .$$

The minimum of L_{ε} is obtained by setting this expression equal to zero,

$$\underline{\varepsilon}^{T} \frac{\partial \underline{\varepsilon}}{\partial \underline{a}_{1}} = 0.$$

From 5), $\frac{\partial \underline{\varepsilon}}{\partial \underline{a}_{1}} = \underline{H}_{s}$ and $\underline{\varepsilon}^{T}\underline{H}_{s} = 0$, or

$$\underline{H}_{s}^{T}\underline{\varepsilon} = 0. \qquad 7)$$

Substituting 5) into 7) gives

$$\underline{H}_{s}^{T} \underline{s} + \underline{H}_{s}^{T} \underline{H}_{s} \underline{a}_{1} = 0$$

or

$$\underline{H}_{s}^{T} \underline{H}_{s} \underline{a}_{1} = -\underline{H}_{s}^{T} \underline{S} . \qquad 8)$$

Note that the matrix $\underline{H}_s^T\underline{H}_s$ and the vector $\underline{H}_s^T\underline{s}$ are equal to

$$\underline{H}_s^T\,\underline{H}_s = \begin{bmatrix} R_{ss}(0) & R_{ss}(1) & \ldots & R_{ss}(q-1) \\ R_{ss}(1) & R_{ss}(0) & \ldots & R_{ss}(q-2) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ R_{ss}(q-1) & R_{ss}(q-2) & \ldots & R_{ss}(0) \end{bmatrix}$$

9)

$$\underline{H}_s^T\underline{s} = \begin{bmatrix} R_{ss}(1) \\ R_{ss}(2) \\ \cdot \\ \cdot \\ \cdot \\ R_{ss}(q) \end{bmatrix} .$$

where $R_{ss}(k) = \sum_{i=0}^{N-1-|k|} s(i)s(i+|k|)$. Equation 8) is a matrix equation representation for the Yule-Walker expressions

$$\sum_{i=1}^{q} a_1(i)R_{ss}(i-k) = -R_{ss}(k), \qquad k = 1, \ldots, q. \qquad 10)$$

If the sequence s(k) is contaminated by additive noise to produce the series x(k)

$$x(k) + s(k) + n(k) , \qquad 11)$$

and an AR(q) model is forced on the noisy data, similar results are obtained. The AR model forced on the noisy data is

$$x(k) = -\sum_{i=1}^{q} a_2(i)x(k-i) + \epsilon(k) \qquad 12)$$

where the $\{a_2(i)\}_1^q$ are the prediction coefficients and $\epsilon(k)$ is the resulting error sequence. If the matrix $\underline{H}_x$ and the vector $\underline{x}$ are formed from the data $x(0), \ldots, x(N-1)$ in a manner similar to $\underline{H}_s$ and $\underline{s}$, then the loss function L_ϵ for the noisy data is

$$L_\epsilon = \sum_{k=0}^{N+q-1} \epsilon^2(k) = \underline{\epsilon}^T\underline{\epsilon}, \qquad 13)$$

with $\underline{\epsilon} = [\epsilon(0) \quad \epsilon(1) \quad \ldots \quad \epsilon(N+q-1)]^T$. Minimizing L_ϵ with respect to the $\{a_2(i)\}_1^q$ results in

$$\underline{H}_x^T\underline{H}_x\underline{a}_2 = -\underline{H}_x^T\underline{x} \qquad 14)$$

as the expression defining the least squares estimate for the $\{a_2(i)\}_1^q$ defined in 12). The elements of the matrix $\underline{H}_x^T\underline{H}_x$ and the vector $\underline{H}_x^T\underline{x}$ are formed from the autocorrelation function of x(k) as in 9) with s(k) replaced by x(k). The $\{a_1(i)\}_1^q$ represent the LPC coefficients determined from the undegraded signal, while the $\{a_2(i)\}_1^q$ are the LPC parameters obtained from noisy data, with no attempt made to eliminate the effects of noise.

Constructing the matrix $\underline{H}_n$ and the vector $\underline{n}$ from the additive noise sequence n(k), the following relationships hold:

$$\underline{H}_x = \underline{H}_s + \underline{H}_n \qquad 15)$$

$$\underline{H}_x^T\underline{H}_x = \underline{H}_s^T\underline{H}_s + \underline{H}_n^T\underline{H}_n + \underline{H}_s^T\underline{H}_n + \underline{H}_n^T\underline{H}_s. \qquad 16)$$

The $\underline{H}_n^T\underline{H}_n$ term is a matrix formed of the autocorrelation terms of n(k) and the terms $\underline{H}_s^T\underline{H}_n$ and $\underline{H}_n^T\underline{H}_s$ contain the crosscorrelation terms between n(k) and s(k). If it can be assumed that s(k) and n(k) are uncorrelated, 16) becomes

$$\underline{H}_x^T\underline{H}_x = \underline{H}_s^T\underline{H}_s + \underline{H}_n^T\underline{H}_n, \qquad 17)$$

With 11), 15), and 17) substituted into 14), we have

$$[\underline{H}_s^T\underline{H}_s + \underline{H}_n^T\underline{H}_n]\ \underline{a}_2 = -[\underline{H}_s + \underline{H}_n]^T(\underline{s} + \underline{n}) = -[\underline{H}_s^T\underline{s} + \underline{H}_n^T\underline{n}] \qquad 18)$$

Where the assumption of uncorrelated signal and noise is used to reduce the right hand side of 18). Solving equations 8) and 18) for $\underline{a}_1$ and $\underline{a}_2$, respectively,

$$\underline{a}_1 = -[\underline{H}_s^T\underline{H}_s]^{-1}\underline{H}_s^T\underline{s} \qquad 19)$$

and

define the least squares estimates for $\underline{a}_1$ and $\underline{a}_2$. The vector $\underline{a}_2$ can be related to $\underline{a}_1$ by premultiplying 18) by $[\underline{H}_s^T\underline{H}_s]^{-1}$ to give

$$[I + (\underline{H}_s^T\underline{H}_s)^{-1}\ \underline{H}_n^T\underline{H}_n]\underline{a}_2 = -[\underline{H}_s^T\underline{H}_s]^{-1}\ \underline{H}_s^T\underline{s} - [\underline{H}_s^T\underline{H}_s]^{-1}\ \underline{H}_n^T\underline{n}$$

$$= \underline{a}_1 - [\underline{H}_s^T\underline{H}_s]^{-1}\ \underline{H}_n^T\underline{n}$$

or

$$\underline{a}_2 = [I + (\underline{H}_s^T\underline{H}_s)^{-1}\ \underline{H}_n^T\underline{H}_n]^{-1}\ \underline{a}_1 - [I + (\underline{H}_s^T\underline{H}_s)^{-1}\ \underline{H}_n^T\underline{H}_n]^{-1}\ [\underline{H}_s^T\underline{H}_s]^{-1}\ \underline{H}_n^T\underline{n}$$

$$= [\underline{H}_s^T\underline{H}_s + \underline{H}_n^T\underline{H}_n]^{-1}\ \underline{H}_s^T\underline{H}_s\underline{a}_1 - [\underline{H}_s^T\underline{H}_s + \underline{H}_n^T\underline{H}_n]^{-1}\underline{H}_n^T\underline{n} \qquad 21)$$

From 21) it is apparent that the addition of n(k) has degraded the $\underline{a}_2$ in two ways:

1) a bias term $[\underline{H}_s^T\underline{H}_s + \underline{H}_n^T\underline{H}_n]^{-1}\ \underline{H}_n^T\underline{n}$ has been subtracted;

2) the relative magnitudes of the $\{a_2(i)\}_1^q$ have been changed due to the matrix multiplying effect of the expression $[\underline{H}_s^T\underline{H}_s + \underline{H}_n^T\underline{H}_n]^{-1}\ \underline{H}_s^T\underline{H}_s$.

The results of equations 18) through 21) are valuable in showing the distortion possible when noise is added to a sequence that is to be the input to an LPC system. These results are based on the explicit assumption that s(k) and n(k) are uncorrelated and fail to account for nonzero crosscorrelation terms (the terms $\underline{H}_s^T\underline{H}_n$, etc.). Results showing the distortion introduced by n(k) on the inverse spectrum derived from the $\{a_2(i)\}_1^q$ and the effects of assuming n(k) and s(k) are uncorrelated will be shown in the section on Preliminary Results.

ARMA Model Approach

Inclusion of the effects of additive white noise upon an AR(q) process is discused in [3], [11], and [14]. The potential advantage of this approach is to include the noise effects explicitly in a more general model than the original AR(q) process. The model is developed on the following assumptions:

1) s(k) is a proper AR(q) sequence described by

$$\sum_{i=0}^{q} a(i)s(k-i) = \varepsilon(k) \qquad 22)$$

for $a(0) = 1$, $a(q) \neq 0$, and $q \geq 1$, with $\varepsilon(k)$ independent, identically distributed (i.i.d.) $N(0, \sigma_\varepsilon^2)$ and S(k) stationary;

2) s(k) is contaminated by n(k) to form the observable data x(k),

$$x(k) = s(k) + n(k), \qquad 23)$$

where s(k) and n(k) are independent and n(k) is i.i.d. $N(0, \sigma_n^2)$.

The model has q+2 parameters--$\{a(i)\}_1^q$, σ_ε^2, and σ_n^2. The data available for analysis to determine estimates of these parameters is the sequence x(0), ..., x(N-1).

Combining 22) and 23), we have

$$\sum_{i=0}^{q} a(i)x(k-i) = \sum_{i=0}^{q} a(i)n(k-i) + \varepsilon(k), \quad a(0) = 1. \qquad 24)$$

A sequence y(k) is defined as

$$y(k) = \sum_{i=0}^{q} a(i)x(k-i) \qquad 25)$$

or

$$y(k) = \sum_{i=0}^{q} a(i)n(k-i) + \varepsilon(k). \qquad 26)$$

If $R_{yy}(k) \overset{\Delta}{=} E[y(i)y(i+k)]$, it can be shown, using 26), that $R_{yy}(k) = 0$ for $|k| > q$. From 26), y(k) is seen to be stationary. Combining this with the property that $R_{yy}(k) = 0$, $|k| > q$, shows y(k) to be a moving average sequence MA(p), with $p \leq q$. Also from 26), $R_{yy}(q) = \sigma_n^2 a(q) \neq 0$, by the hypothesis under assumption 1) above. As a result, y(k) is a MA(q) process, and there exists a sequence of random variables $\eta(k)$, i.i.d. $N(0,\sigma_\eta^2)$ and constants $\{b(i)\}_1^q$ such that

$$y(k) = \sum_{i=0}^{q} b(i)\eta(k-i), \quad b(0) = 1. \qquad 27)$$

Combining 25) and 27) gives

$$\sum_{i=0}^{q} a(i)x(k-i) = \sum_{i=0}^{q} b(i)\eta(k-i), \qquad a(0) = b(0) = 1 \qquad 28)$$

and the sequence x(k) can be viewed as an ARMA(q,q) process. While the original model has q+2 parameters--$\{a(i)\}_1^q$, σ_ε^2, and σ_n^2--the new model has 2q+1 parameters--$\{a(i)\}_1^q$, $\{b(i)\}_1^q$, and σ_η^2. From 27),

$$R_{yy}(k) = \sigma_\eta^2 \sum_{i=0}^{q-|k|} b(i)b(i+k), \qquad b(0) = 1, \qquad 29)$$

so the expanded parameter set could equivalently be expressed as $\{a(i)\}_1^q$ and $\{Ryy(i)\}_0^q$.

Using the definition for y(k) in 26),

$$R_{yy}(k) = \sigma_\varepsilon^2\delta(k) + \sigma_n^2 \sum_{i=0}^{q-|k|} a(i)a(i+k), \qquad a(0) = 1, \qquad 30)$$

where $\delta(k) = \begin{cases} 1, & k = 0 \\ 0, & k \neq 0 \end{cases}$.

Thus, the addition of n(k) to s(k) produces the following relationships between the parameters:

1) equation 29) gives the autocorrelation function Ryy(k) for any MA(q) process;

2) another definition for Ryy(k) given in 30) arises as a result of the noise model defined by 22) and 23);

3) the ARMA parameters $\{a(i)\}_1^q$ and $\{b(i)\}_1^q$ for the process x(k) are related through the autocorrelation function $R_{yy}(k)$, the relationship being expressed by 29) and 30).

A comparison of the ARMA model approach just described with a forced LPC fit of the data, represented by the solution of 20), shows two interesting facts. First, the forced LPC model, from a spectral point of view, must match the spectral characteristics of the input x(k) as closely as possible. This spectral match includes those characteristics introduced by the noise. The next section will present examples illustrating the flattening effect white noise has on a forced LPC fit. The second observation involves the assumption of the model form. If the original sequence s(k) is AR(q), then the addition of white noise results in an ARMA(q,q) process, x(k). This process is equivalently an AR(∞) process. The forced LPC fit is actually representative of the first step in the process discussed in [5] for estimating ARMA parameters, that is, underfitting the AR(∞) process. The ARMA model approach can then be viewed as a procedure by which the AR(q) and MA(q) parameters are estimated from the AR(∞) parameters.

Having identified the mathematical relationships and properties for the ARMA approach to parameter estimation in the presence of noise, the next phase describes the procedure for estimating the AR parameters of the original signal s(k). The first step requires the estimation of the ARMA parameters of the expanded model--$\{\bar{a}(i)\}_1^q$, $\{\bar{b}(i)\}_1^q$, and $\bar{\sigma}_n^2$--the symbol "_" indicating estimate. This represents the identification of the series x(k) according to 28). This set of parameters is then transformed to the equivalent set--$\{\bar{a}(i)\}_1^q$ and $\{\bar{R}_{yy}(i)\}_0^q$--by the relationship defined in 29). At this stage there are estimates for the original AR parameters $\{a(i)\}_1^q$. The efficiency of these estimates will depend upon the technique used to estimate the parameters of an ARMA process. The estimation procedure, however, has not yet used the information available under assumption 2) of the model. This information is carried in the relationship given by 30) between $\{R_{yy}(i)\}_0^q$, on one hand, and $\{a(i)\}_1^q$, σ_ε^2, and σ_n^2, on the other.

Using nonlinear regression theory, refinements are made on the estimates for the original parameter set according to

$$\underline{z} = \underline{f}(\underline{\theta}) + \underline{e}, \tag{31)}$$

where $\underline{z} = [\bar{a}(1)\ \ \bar{a}(2)\ \ \ldots\ \ \bar{a}(q)\ \ \bar{R}_{yy}(0)\ \ \bar{R}_{yy}(1)\ \ \ldots\ \ \bar{R}_{yy}(q)]^T$

and $\underline{\theta} = [\ \tilde{a}(1)\ \ \tilde{a}(2)\ \ \ldots\ \tilde{a}(q)\ \ \tilde{\sigma}_\varepsilon^2\ \ \tilde{\sigma}_n^2\ ]^T$. The $\{\bar{a}(i)\}_1^q$ and $\{\tilde{a}(i)\}_1^q$ in $\underline{z}$ and $\underline{\Theta}$, respectively, are estimates for the AR parameters. The $\{\bar{a}(i)\}_1^q$ result from the ARMA parameter estimation step. Then the nonlinear regression stage produces the $\{\tilde{a}(i)\}_1^q$. In 31) the nonlinear functional relationship $\underline{f}(\cdot)$ represents that of 30), and $\underline{e}$ is a (2q+1) x 1 vector reflecting the error between the ARMA parameter estimates found in $\underline{z}$ and the theoretical relationship to these estimates represented by 30). $\underline{f}(-)$ maps from the original q+2 parameter set to the 2q+1 parameter set of the expanded model. The nonlinear regression technique attempts to find the estimate of $\underline{\theta}$ which will minimize $\underline{e}$ in the least squares sense. The Gauss-Newton method is proposed for this task [11] and will be discussed in the next part.

Algorithms

In developing the ARMA approach in the previous part, two steps require major algorithms: the estimation of ARMA parameters from a time series and a nonlinear regression technique. The nonlinear regression (NLR) will be presented first. Equation 31) describes the nonlinear relationship between the parameter sets $\underline{z}$ and $\underline{\theta}$. The 2q+1 equations comprising 31) can be written as

$$z_t = f_t(\underline{\theta}) + e_t, \quad t = 1, \ldots, 2q+1. \qquad 32)$$

The metric [8] for evaluating the effectiveness of $\underline{\theta}$ in minimizing $\underline{e}$ is given by

$$Q(\underline{\theta}) = \sum_{t=1}^{2q+1} [z_t - f_t(\underline{\theta})]^2. \qquad 33)$$

Using 30) to define the $f_t(\cdot)$, t = 1, ..., 2q+1, gives the following set of equations:

$$\bar{a}(i) = \tilde{a}(i) + e_i, \quad i = 1, \ldots, q$$

$$\bar{R}_{yy}(0) = \tilde{\sigma}_\varepsilon^2 + \tilde{\sigma}_n^2 \sum_{i=0}^{q} \tilde{a}^2(i) + e_{q+1}, \qquad 34)$$

$$\bar{R}_{yy}(k) = \tilde{\sigma}_n^2 \sum_{i=0}^{q-|k|} \tilde{a}(i)\tilde{a}(i+k) + e_{q+k+1}, \quad k = 1, \ldots, q,$$

where $\tilde{a}(0) = 1$. The $\{\tilde{a}(i)\}_1^q$, $\tilde{\sigma}^2$, and $\tilde{\sigma}_n^2$ are chosen to minimize $Q(\underline{\theta})$,

$$Q(\underline{\theta}) = \sum_{i=1}^{2q+1} e_i^2. \qquad 35)$$

An iterative procedure based on the Gauss-Newton method or modified Gauss- Newton method will yield a solution $\underline{\Theta}$ to 32) having the properties of convergence for a finite number of functional relationships $f_t(\cdot)$, and the $\underline{\Theta}$ will be asymptotically efficient [7]. The Gauss-Newton method is based on the linearization of the nonlinear functions $f_t(\cdot)$ about the solution $\underline{\Theta}$.

The second of the major algorithms to be discussed is the method for obtaining estimates for the ARMA parameters of a time series. The method used will probably be based on one of following techniques: Hannan [6]; Graupe, Krause, and Moore [5]; or Steiglitz [12]. Hannan's technique is a three step procedure adaptable to iteration if desired. The following is a brief summary of the procedure. Notation is based on that used previously in this section.

1) Initial estimates for the AR parameters. Using the property that $R_{yy}(k) = 0$ for $|k| > q$ for a MA(q) process y(k), estimates of the AR coefficients $\{a(i)\}_1^q$ are found from the solution of

$$\sum_{i=1}^{q} \bar{a}(i)R_{xx}(i-k) = -R_{xx}(k), \quad k = q+1, \ldots, 2q.$$

with $$R_{xx}(k) = \sum_{i=0}^{N-1-|k|} x(i)x(i+k) \quad , \quad k = 0, \ldots, 2q.$$

2) Initial estimates for the MA parameters. Form $\bar{R}_{yy}(k) = \sum_{i=0}^{q} \sum_{j=0}^{q} a(i)a(j)R_{xx}(k+i-j)$. 36) Note that this is the autocorrelation function of y(k) obtained by using 25). If the power spectrum estimate for y(k), $S_{yy}(\omega)$, is non-negative, it can be factored to give $\{\bar{b}(i)\}_1^q$, estimates of the MA parameters. This is essentially the Autocorrelation Partial Realization method of Atashroo [1]. If $\bar{S}_{yy}(\omega) \not\geq 0$, $0 \leq \omega \leq \pi$, then Hannan's technique proceeds to obtain the $\{\bar{b}(i)\}_1^q$ through several intermediate steps.

3) Refine the initial ARMA estimates. The initial estimates $\{\bar{a}(i)\}_1^q$ and $\{\bar{b}(i)\}_1^q$ are modified by calculating correction factors which are added to the initial estimates found in step 2) above. The procedure is to first modify the $\{\bar{a}(i)\}_1^q$, then the MA parameters $\{\bar{b}(i)\}_1^q$, followed by a final

modification of the $\{\bar{a}(i)\}_1^q$. Step 3) can be repeated to form an iterative approach.

The iterative procedure of Step 3) is associated with the solution of equations 29) and 36), while the Gauss-Newton NLR procedure is associated with the solution of 30). Hannan's technique is based upon Fourier transformation of the data and carries a large computational burden.

The technique presented by Graupe, et. al., [5], has the advantage of using only linear operations. The process is summarized as follows:

1) Given an ARMA(q,p) process with AR parameters $\{a(i)\}_1^q$ and MA parameters $\{b(i)\}_1^p$, consider this to be an AR(∞) process with parameters $\{\partial(i)\}_1^\infty$. From the data, estimate the $\{\partial(i)\}_1^N$ for some large integer value for N.
2) Using the expressions generated by the relationship between the $\{\partial(i)\}_1^N$, $\{a(i)\}_1^q$, and $\{b(i)\}_1^p$, a process is obtained for estimating first the $\{b(i)\}_1^p$ and then the $\{a(i)\}_1^q$. Both stages involve the solution of a system of linear equations.

This method is computationally more appealing than Hannan's method. However, the value of N, which determines the order of the $\partial(i)$ approximation, may be rather large for some processes in which the zeros of the MA filter are near the unit circle.

Steiglitz's [12] method for identifying the AR and MA parameters of a process is based on the mode 1 iterative scheme for system identification by Steiglitz and McBride [13]. Steiglitz's approach to identifying the ARMA parameters is the following:

1) Assume the sequence to be identified, x(k), is the output of the unknown system.

2) Assume the input to this system, u(k), is a Kronecker delta function.

3) Minimize the error criterion

$$L_s = \sum_{k=0}^{N-1} e^2(k) = \frac{1}{2\pi j} \oint \left[\frac{A(z)}{A_0(z)} X(z) - \frac{B(z)}{A_0(z)} U(z) \right]^2 \frac{dz}{z} \qquad 37)$$

by the appropriate choice of A(z) and B(z), the z-transforms of the denominator parameters $\{a(i)\}_0^q$ and numerator parameters $\{b(i)\}_0^p$, respectively. U(z) and X(z) are the z-transforms of the unknown system's input and output sequences, respectively. $A_0(z)$ is the z-transform of an initial-guess for the denominator filter.

4) By replacing $A_0(z)$ with the A(z) determined in step 3), new estimates for A(z) and B(z) can be found. This iterative procedure can then be continued until the desired accuracy is obtained.

The solution of 37) for the $\{a(i)\}_1^q$ and the $\{b(i)\}_0^p$ involves the solution of a system of q+p+1 linear equations.

Software

The last part of this section describes the software needed to generate the data base used for simulating an AR process corrupted by additive white noise. The data derives from two noise files obtained by digitizing the output of an analog noise generator. The analog signal is prefiltered with a low pass filter having a 3.2 kHz cutoff frequency and is sampled at 6667. Hz. From two files generated in this manner, one is scaled so that its sample variance is approximately 1.0. This sequence is $\varepsilon(k)$, the excitation sequence for the AR process to be simulated. The second noise file $n_0(k)$ is used to generate $n(k)$, the additive noise.

A two step process generates the data base for a particular AR process.

1) Design and generation of the AR process. Using the program ARPGEN.SAV, the user designs an AR(q) process, $1 \geq q \geq 20$. The filter is checked for stability by analyzing the partial correlation coefficients derived from the AR parameters. The power spectrum of the inverse filter corresponding to the AR coefficients is calculated and displayed. If the model is acceptable, the time series $s(k)$ corresponding to this process is computed using the noise sequence $\varepsilon(k)$ mentioned above as the excitation to the AR filter.

2) Generation of the additive noise. After finding approximations to the variances of s(k) generated in 1) and the noise file $n_0(k)$ described above, the program SPLUSN.SAV can be used to generate either of the outputs

$$n(k) = c \cdot n_0(k)$$

or

$$x(k) = s(k) + c \cdot n_0(k).$$

The constant c is computed by SPLUSN.SAV so that the quantities $c \cdot n_0(k)$ and s(k) will have a specified signal-to-noise ratio.

Using the above system of data and the two programs discussed, various AR processes and additive noise sequences can easily be synthesized in preparation for analysis by the ARMA noise model approach.

PRELIMINARY RESULTS

One of the objectives of this research is to characterize the effects of additive white noise on LPC analysis systems. The following data illustrate the degradation caused by additive noise. Results are presented for a speech waveform sample at varying levels of noise. Figure 1 shows the speech frame used as the example for this section. The time waveform is shown in Fig. 1a). Sampled at 6667. Hz, this frame of 128 samples corresponds to about 19 msec. of speech. This frame represents a portion of the schwa vowel /ə/ in the word "rust". This particular vowel was selected because of the nearly uniform distribution of formants. Also, the formants drop in peak magnitude at a constant rate as frequency increases (on a dB scale). Fig. 1b) shows the DFT of this frame of speech, after windowing with a Hamming window. On the dB scale, the nearly uniform formant structure of the schwa vowel is apparent. Superimposed on Fig. 1b) is the spectrum corresponding to a 10 pole LPC fit of this frame. The LPC spectrum is smoother and matches the formant peaks well.

Fig. 2, a)-e), show the effects of additive white noise with progressively smaller signal-to-noise ratios: 40, 30, 20, 10, and 0 dB. The SNR is found by averaging the energy in the speech and the noise sequences over several seconds. The ratio of these energies is then used to determine the SNR, defined as

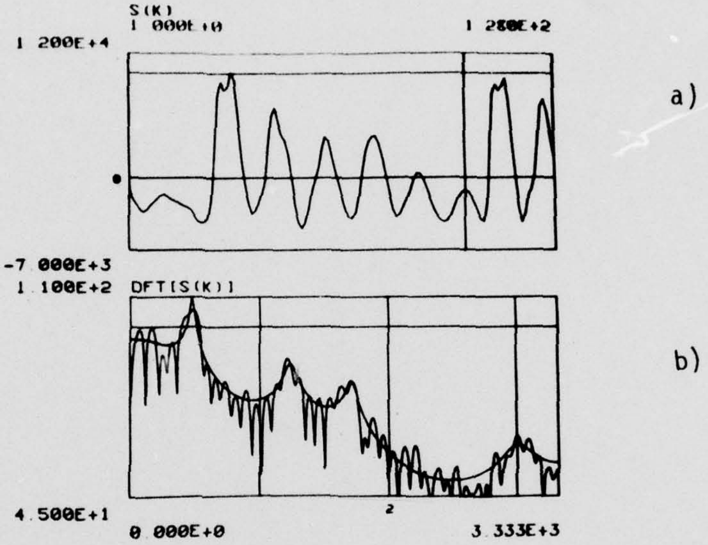


Figure 1: Example frame used as s(k)
a) 128 samples of the vowel /ə/, sampled at 6667. Hz.
b) Spectrum of /ə/ and a 10 pole LPC fit to that spectrum.

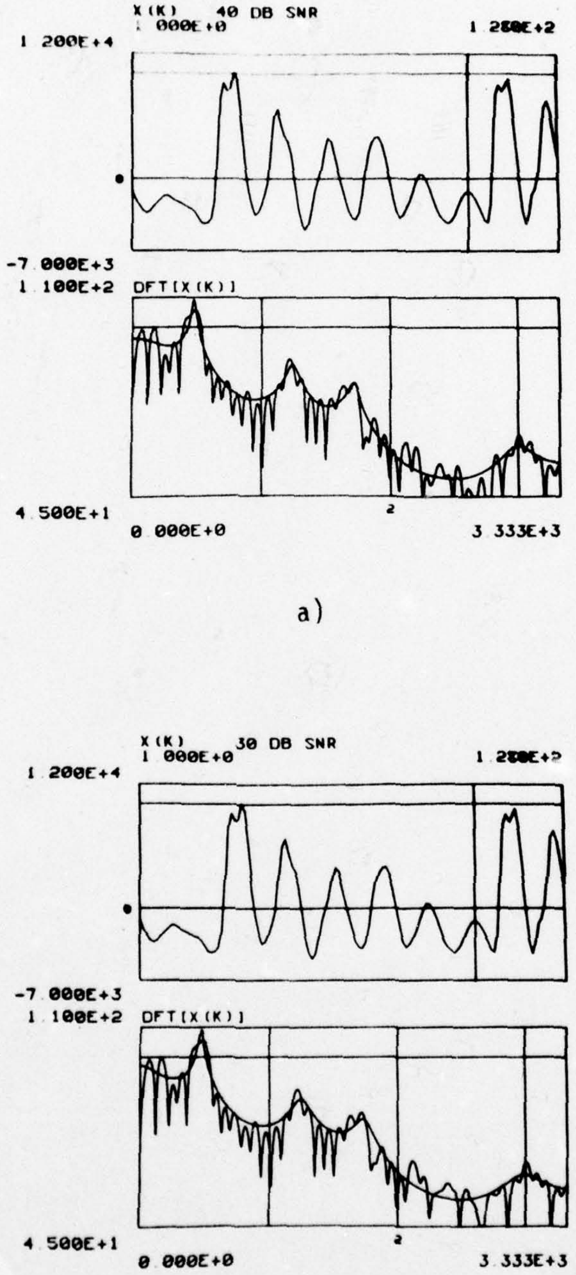
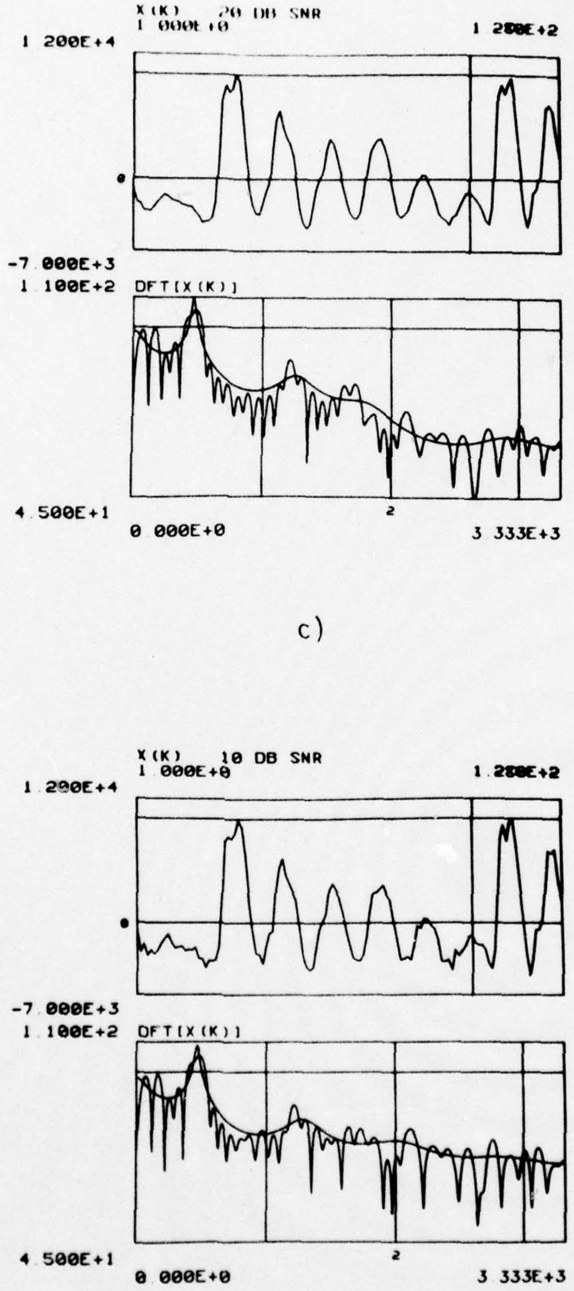


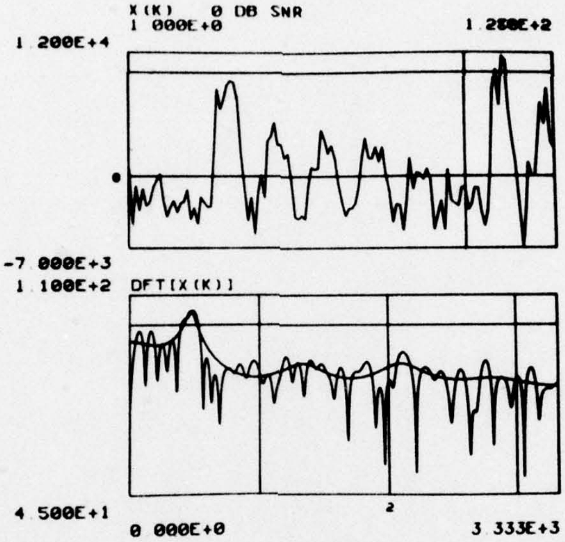
Figure 2: Illustrations of the effects of additive white noise on the example speech frame and 10 pole LPC approximations to the resulting spectrum.
a) 40 dB SNR
b) 30 dB SNR



c)

d)

Figure 2: c) 20 dB SNR
d) 10 dB SNR



e)

Figure 2: e) 0 dB SNR

$$SNR = \frac{\sum s^2(k)}{\sum n^2(k)} .$$

Superimposed on each spectral plot is the corresponding 10 pole LPC fit. All spectral graphs in Figures 1 and 2 are on the same scale and can be compared directly. The following noise effects are noted:

1) with decreasing SNR, the noise "floor" rises, obscuring more of the formant structure of the speech;
2) the formants identified by LPC analysis in increasingly poorer SNR's tend to be wider in bandwidth and have their peaks at slightly higher frequencies;
3) the formant structure identified by LPC is badly degraded for SNR's below about 20 dB.

The importance of the assumption of uncorrelated signal and noise is demonstrated in the next set of data. This assumption is primary to the autocorrelation correction methods of parameter estimation [1], [4]. Figure 3a) shows R_{ss} (k), the autocorrelation function for the frame of speech being discussed. Plotted in Fig. 3b) are R_{nn} (k), the noise autocorrelation function, and R_{sn} (k), one of the crosscorrelation functions. The abscissa in Fig. 3a) and b) starts at lag k = 0 and is followed by 100 lags for k = 1, ..., 100. The last 100 points are the negative lags in the order k = -100, ..., -1. The noise used for Fig. 3

corresponds to a 10 dB SNR. It is obvious that $R_{sn}(k) \neq 0$, based on the estimation of $R_{sn}(k)$ from

$$R_{sn}(k) = \sum_{i=0}^{N-1} s(i)n(i+k). \tag{38}$$

The spectral implications of this are shown in Fig. 3c), which shows four spectral curves determined from LPC coefficients calculated from the four autocorrelations:

i) $R_{ss}(k)$,

ii) $R_{ss}(k) = R_{xx}(k) - R_{nn}(k) - R_{sn}(k) - R_{ns}(k)$

iii) $\hat{R}_{ss}(k) = R_{xx}(k) - R_{nn}(k)$,

iv) $R_{xx}(k)$.

Note that i) and ii) result in the same spectral plot. The explicit assumption of uncorrelated signal and noise is used in iii), while iv) corresponds to LPC coefficients determined from noisy data, with no correction attempted. Fig. 3c), curve iii), shows the inadequacy of the uncorrelated assumption for the autocorrelation correction modeling approach. Even though curve iii) appears superior to iv), in a large percentage of frames, the LPC algorithms will fail, producing unstable inverse filters, when the autocorrelations of $R_{ss}(k)$ are based on iii). An autocorrelation matrix which is not positive definite causes this.

The preceding results show LPC analysis procedures are sensitive to additive white noise degradation. The autocorrelation correction method, while appealing, is not

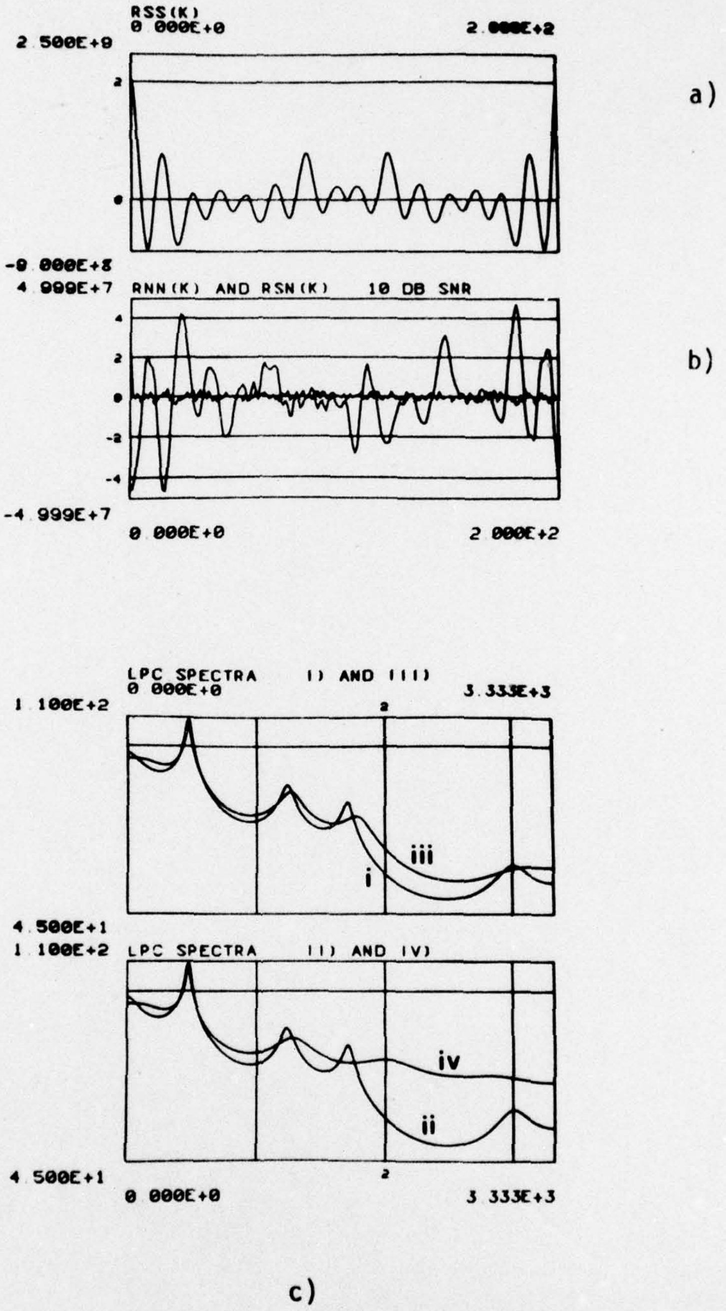


Figure 3: Comparison of autocorrelation and crosscorrelation sequences for the sample frame with a 10 dB SNR.

a) $R_{ss}(k)$

b) $R_{nn}(k)$ and $R_{sn}(k)$

c) 10 pole LPC spectra based on autocorrelations i-iv discussed in text.

successful if the uncorrelated signal-noise assumption is made.

Preliminary Results of the ARMA Model for Additive White Noise

In developing the ARMA model suggested for estimating AR parameters when noise is present, the major components of the procedure suggested by Pagano have been implemented. Initial tests of the algorithm performed on speech samples were unsuccessful in either proving or disproving the technique or verifying the operation of the software. It is this result that led to the used of synthetic AR processes for testing of this approach. This will avoid the critical assumption that the original undegraded time series, speech in this case, can be modeled as an AR(q) process. It also led to a perturbation analysis as a way of verifying the correct operation of the nonlinear regression (NLR) software. The perturbation study of the NLR technique also provides information about the ability of NLR to converge to the correct parameter set when the initial parameters are in error. The perturbation analysis is based on the following:

1) A set of q AR parameters and variances σ_ε^2 and σ^2 are selected to characterize an AR process in additive noise.
2) Using equation 30) of the previous section, the $\{R_{yy}(k)\}_0^q$ are calculated from the q+2 parameters

selected in step 1).

3) With $\underline{z} = \underline{f}(\underline{\theta}) + \underline{e}$ describing the nonlinear mapping from $\underline{\theta}$ to $\underline{z}$, construct $\underline{z}$ from the $\{a(i)\}_1^q$ and $\{R_{yy}(k)\}_o^q$.

4) The parameters of $\underline{\theta}$ are the $\{a(i)\}_1^q$, σ_ε^2 and σ_n^2. The perturbation will be introduced into $\underline{\theta}$.

5) Prior to entering the NLR routine, some or all of the parameters of $\underline{\theta}$ are perturbed.

6) The NLR iterative procedure then attempts to correct $\underline{\theta}$ so that $\underline{e} \rightarrow 0$.

For the q = 1 case, an analytical development of the NLR technique is possible, and the results can be used to predict the findings of the computer analysis.

The following points can be deduced from the theoretical study:

1) For errors in either or both of the σ_ε^2 and σ_n^2 parameters of $\underline{\theta}$, the errors are corrected in one iteration without introducing errors in any other component of $\underline{\theta}$.

2) Any error in the single AR parameter a(1), whether accompanied by errors in σ_ε^2 or σ_n^2, or not, is corrected in one iteration. The first iteration, however, produces an error in the σ_ε^2 component of $\underline{\theta}$. It thus requires one additional iteration to correct that error.

3) If the perturbation of the a(1) parameter results in an initial value of a(1) = 0, then the

theoretical development predicts that the NLR technique based on the Gauss-Newton method will fail on the first iteration.

Computer simulation of the perturbation analysis for the q = 1 case verified the above theoretical results with the following qualification: the NLR Gauss-Newton technique involves the inversion of a matrix which becomes ill-conditioned as σ_n^2, the actual noise variance, or $\tilde{\sigma}_n^2$, the estimate of that variance, get large. This observation results from the finite word length and round-off effects associated with computers. This is not predicted by the theoretical analysis, though it is apparent from the matrix equations associated with this technique why this would happen. Its presence in the simulation is the justification for using small variance sequences for $\varepsilon(k)$ and $n(k)$-- on the order of 100. or less. To prevent the increase of the effects of quantization noise, data must be stored on disk in the unpacked format of 128 data points per disk block.

Tables I and II show perturbation effects for the q = 1 case for the parameters listed in the tables. The examples in these tables represent the worst case in which there are errors in all three parameters. For demonstration purposes, the errors are +200% for each parameter. The value of $\underline{\theta}$ at each iteration is given to four decimal places.

Table I

Perturbation Analysis, q = 1

	a(1)	σ_ε^2	σ_n^2	Dist
Design Parameters	0.8	1.0	1.95	-
Initial Estimates	2.4	3.0	5.85	2.56
Iteration #1	0.8000	18.3679	4.5499	7.52×10^{-11}
#2	0.8000	1.0000	1.9500	8.44×10^{-14}

Table II

Perturbation Analysis, q = 1

	a(1)	σ_ε^2	σ_n^2	Dist
Design Parameters	-0.8	1.0	3.03	-
Initial Estimates	-2.4	3.0	9.09	2.56
Iteration #1	-0.8000	27.9869	7.0699	2.44×10^{-10}
#2	-0.8000	1.0000	3.0301	3.09×10^{-11}

Another more interesting perturbation study is shown in Table III. In this case, the AR coefficients are a(1) = -2.3, a(2) = 2.4, a(3) = -1.6, and a(4) = 0.6. For purposes of demonstration, the initial estimates are $\bar{a}(i) = -a(i)$, a -200% error in estimating each coefficient. The actual variances and their estimates are shown in Table III, which illustrates the convergence of $\underline{\theta}$ to the correct solution. From the table, it is evident that, to four decimal places, the convergence for the four AR coefficients is complete in 10 iterations. Note that in Tables I, II, and III, the entry for "Dist" under each iteration is the l_2 distance between the $\{a(i)\}_1^q$ and the $\{\tilde{a}(i)\}_1^q$. The noise variance σ_n^2 in each of the three perturbation examples corresponds to a 0 dB SNR when compared to the sample variance of the respective AR process.

While the preceding discussion demonstrates the power of the NLR technique, using a Gauss-Newton approach, the following is an initial attempt at applying the entire procedure to the synthetic AR(1) sequences. The results seem to indicate two major problems:

1) The initial estimates for the {a(i)} are poor. Hannan's method begins by estimating the AR parameters by

$$\sum_{i=1}^{q} a(i)R_{xx}(i-k) = - R_{xx}(k), \quad k = q+1, \ldots, 2q. \qquad 39)$$

The next step in Hannan's procedure is to calculate the approximate autocorrelation function of the moving average sequence y(k):

$$\bar{R}_{yy}(k) = \sum_{i=0}^{q} \sum_{j=0}^{q} \bar{a}(i)\bar{a}(j)R_{xx}(k+i-j), \quad k = 0, \ldots, q. \qquad 41)$$

If the spectrum $\bar{S}_{yy}(\omega)$ corresponding to $\bar{R}\,yy(k)$ is non-negative, the spectrum can be factored to give the $\{b(i)\}_1^q$, initial estimates of the MA parameters. This spectral factorization represents the solution of

$$\bar{R}_{yy}(k) = \bar{\sigma}_n^2 \sum_{i=0}^{q-|k|} \bar{b}(i)\bar{b}(i+k), \quad \bar{b}(0) = 1, \qquad 41)$$

for the $\{\bar{b}(i)\}_1^q$ and $\bar{\sigma}_n^2$. This solution is nonlinear in nature. The procedure proposed by Hannan then uses an iterative stage to improve these initial estimates. Presently, the only steps of this technique being used are the calculation of the $\{\bar{a}(i)\}_1^q$ and the $\{\bar{R}_{yy}(k)\}_0^q$ using equations 40) and 41). This would avoid the nonlinear spectral factorization necessary to obtain the $\{\bar{b}(i)\}_1^q$ and $\bar{\sigma}_n^2$, which are then recombined by 41) to obtain the $\{\bar{R}_{YY}(k)\}_0^q$ for the Gauss-Newton NLR routine. However, further improvement on these parameters may be necessary to prevent the NLR technique from diverging.

SECTION IV

MULTIRATE SIGNAL PROCESSING

H. Ravindra

Abstract

The aim of this project is to simulate a system on a Digital Computer, which can increase or decrease the sampling rate of an acoustic signal. These two operations are called Interpolation and Decimation respectively. This project is the first phase of a larger project which involves the simulation of a CVSD system, with an idea of studying the problems of tandeming. Since the CVSD performance (signal-to-noise ratio) is better at higher sampling rates, an Interpolation/Decimation scheme is required to translate the sampling rate from the lower 6.67 KHz to higher rates.

Introduction

In many digital signal processing systems, e.g., vocoders, modulation systems and digital waveform coding systems, it is necessary to alter the sampling rate of a digital signal. In the present context, we are interested in the last application. The multirate signal processing system is actually the first phase of a larger project. The main project involves studying the problems of tandeming a CVSD system with a vocoder. This requires simulation of two systems on a digital computer. They are, the multirate signal processing system and a CVSD encoder/decoder system. In the present write up, the multirate signal processing system is described. Firstly, interpolation and decimation of the sampling rate of a digital signal are shown to be simple linear filtering processes. Later, it will be shown that an interpolator or a decimator can be implemented optimally over several stages. Also, the suitability of FIR filters over IIR filters for this application will be discussed. In the present work, both non-optimal (single-stage) and optimal (multi-stage) interpolators and decimators have been implemented and comparative results are presented at the end of the write up.

I. Interpolation and Decimation as Linear Filtering Processes

Let $\hat{x}(t)$ be a continuous time signal and $\hat{x}(n)$ the sampled version of $\hat{x}(t)$

i.e.,

$$x(n) = \hat{x}(nT) \text{ where T is the sampling period.}$$

It can be shown that the Fourier Transform of $\hat{x}(t)$ and $x(n)$ are related as follows:

$$X(e^{j\omega T}) = \frac{1}{T} \sum_{k=-\infty}^{\infty} \hat{X}(\omega + k\frac{2\pi}{T})$$

If $\hat{x}(t)$ is band limited, i.e., $\hat{x}(\ \omega\ ) = 0$ for $|\omega| \geq \Omega$ and if $T \leq \frac{\pi}{\Omega}$ (to avoid aliasing), then

$$X(e^{j\omega T}) = \frac{1}{T} X(\omega) \quad \frac{-\pi}{T} \leq \omega \leq \frac{\pi}{T}.$$

(a) **Sampling rate reduction by integer factors:**

Suppose that the desired sampling period is $T' = MT$. If M is an integer, then,

$$\begin{aligned} y(n) &= \hat{x}(nT') = \hat{x}(nMT) \\ &= x\ (Mn). \end{aligned}$$

So, decimation by an integer factor M is achieved simply by "picking off" every Mth sample from the original signal sampled at a rate of $\frac{1}{T}$. Before applying the decimation process, it may be necessary to perform low pass filtering of the original signal x(n), to avoid aliasing, as shown below.

It is seen that the Fourier Transform of the decimated signal is,

$$Y(e^{j\omega T'}) = \frac{1}{M}\sum_{\ell=0}^{M-1} X\left(e^{j(\omega T'-2\pi\ell)/M}\right)$$

so, unless $T' \leq \frac{\pi}{\Omega}$ (i.e., $T \leq \frac{\pi}{M\Omega}$), aliasing results and it is not possible to recover the original signal from the sampled (decimated) version. Hence, prefiltering (lowpass) is needed when $T' > \pi/\Omega$ (i.e., $T > \frac{\pi}{M\Omega}$) to avoid aliasing. The low pass filter must have a cutoff frequency of $\pi/M\Omega$.

(b) <u>Sampling rate increase by integer factors:</u>

Let T' = T/L. If L is an integer, the new sampling rate 1/T' would be equal to the original rate 1/T multiplied by the factor L. It has been shown that interpolation involves two steps. In the first step, L-1 zero samples are introduced between samples of the original signal and in step 2, the resulting signal is lowpass filtered.

The Fourier transform of the original signal padded with zeros can be shown to be,

$$Y(e^{j\omega T'}) = X(e^{j\omega T'L}) = X(e^{j\omega T})$$

and it is seen that it is periodic with a period of $2\pi/T$. But the properly interpolated signal should be periodic with a period of $2\pi/T'$. Therefore, low pass filtering is needed, after padding with zeros, to keep only the base band and attenuate the inner lobes. Also, it is clear that the passband gain of the filter should be equal to the interpolation ratio.

(c) Changing by non-integer factors:

Let the new sampling period be $T' = \frac{M}{L}T$, where M and L are integers. An interpolation or decimation by a non-integer factor can be realized by first interpolating by the factor L, and then decimating by a factor M. If the overall factor by which the rate is changed is less than unity, LP filtering is needed before the decimation step to avoid aliasing.

II. Selection of the Type of Filter:

The choice is between FIR and IIR filters. The following advantages of the FIR filter make this class more suitable then the IIR filters.

(i) FIR filters can be realized with precisely linear phase and can be made to approach the ideal magnitude response by reducing the stop band and passband ripples, and also by reducing the transition bandwidth. Of course, this requires long filters. Though IIR filters can be realized with extremely good magnitude response, they suffer from non-linear phase characteristics.

(ii) It can be shown that the post filtering required for interpolation and prefiltering required for decimation can be combined to form a single filter. This filter accepts zero padded signal and generates the final output - (Interpolated and decimated, to realize a non-integer change in rate). If FIR filters are used, the fact that the filter sees one nonzero sample in every L samples and produces an output sample in every M samples, results in reduced computational complexity. Also, during the decimation process, the symmetry of the filter can be utilized to reduce computation by a factor of 2.

III. Optimal Design of Interpolators and Decimators

Crochiere and Rabiner have shown that the post filter required in the interpolation step and prefilter required in the decimation step, can be combined into a single filter to realize a non-integer ratio of change of rate. Also, they show that the amount of computation can be reduced considerably by implementing interpolators and decimators in a multistage configuration, and develop techniques for their optimal design.

(a) The physical reason for reduced computation when multistage implementation is used, is the following.

Since the FIR filter accepts a zero padded signal, it sees one nonzero sample in every L samples. Also since it is performing predecimation low pass filtering, it need output one sample in every M samples. Therefore, the computational complexity (number of multiplies and adds: MADS) is proportional to N/(LM). That is, the number of MADS required to generate each output point is N/(LM). It can be seen from Figure 4 that a multistage implementation of a decimator requires the longest filter for the last stage and relatively very short filters for the earlier stages. The opposite is true of interpolators. For the same end results, the ripple specifications on each state is more severe (by a factor equal to number of stages) than in the case of a single stage implementation. But it is shown that this affects the total computation by a small amount. Since the filter in the last stage (longest) of a multistage decimator and the filter required by a single stage decimator compare as shown in Figures 1 and 2, it is seen that the single stage filter is much longer. So, by realizing a large amount of decimation over the earlier stages and a small amount (so that both L and M are large for the last stage) over the last stage, the number of MADS can be reduced considerably. Very large reduction in the number of MADS required, is realized when the interpolation or decimation ratio is greater than about 20.

(b) Design procedure (optimal decimator)

The number of MADS required per second is shown to be,

$$R_T = D_\infty(\frac{\delta p}{K}, \delta s) f_{ro}\ S(\text{MADS})$$

where, $D_\infty(\frac{\delta p}{K}, \delta s)$ is a function of stopband (δs) and passband (δp) variances, and the number of stages, K. f_{ro} is the initial sampling rate and S(MADS) is a function of the decimation ratios of the K stages, and the transition bandwidth.

Crochiere and Rabiner have shown that D_∞ is not very sensitive to the number of stages (Reference: Table 1), but the function S(MADS) is. So, they have developed various design curves which can be used to find the optimum number of stages and an optimal set of values for the decimation ratios for the K stages. For a two stage design, solution is available in closed form as

$$D_{1OPT} = \frac{2D(1 - \sqrt{D\Delta f/(2-\Delta f)})}{2 - \Delta f(1 + D)}$$

and

$$D_{2OPT} = D/D_{1OPT}, \text{ where } \Delta f = \frac{fs - fp}{fs}$$

and D = required decimation ratio

The design procedure for a general K stage decimator follows.

The specifications are δp, δs, Δf, D and f_{ro}. It has been shown that the use of more than four stages will only increase the complexity of implementation and will not reduce computation any further. Therefore, the values of D_∞ for the specified values of δp, δs are found for K = 1,2,3 and 4 from Table 1. Referring to Figure 5, the value of S, for the specified values of D and f, is found for each value of K=1 through 4. The value of R_T is then computed for each K. Obviously, the value of K, which results in minimum value for R_T, is the optimal value. From the graphs in Figure 6, the decimation ratio for each stage can be found from the specified values for D and Δf.

The same design curves may be used for the design of an optimal interpolator as the processes of interpolation and decimation are duals to each other.

IV. Practical Consideration in the Implementation of Multistage Decimator and Interpolators:

An implementation strategy is described by Crochiere and Rabiner which automatically takes care of the presence of L-1 zero samples between non-zero samples in the input, without actually checking for zeros and also generates only one output point for every M samples.

If the length of the unit sample response of the LP filter, N, is chosen such that

N = QL (where L in the decimation or interpolation ratio and Q is an integer),

then, exactly Q non-zero samples of the input sequence (effectively padded with zeros) are spanned by the unit sample response.

Then, the output sequence is given by

$$y(n) = \sum_{k=0}^{Q-1} h(kL+(nm) \oplus L)x([\frac{nm}{L}] - k)$$

where h(n) is the filter impulse response, $(\ ) \oplus L$ implies the quantity in parentheses modulo L and $[\frac{nm}{L}]$ is the integer value of nM/L. From this, it is seen that the input sequence is to be sequentially addressed for Q of its values, to generate one output point. Also, the filter unit sample response must be addressed by $(KL+(nM) \oplus L)$. But, if h(n) is stored in a scrambled order: h(0), h(L),...., $h((Q-1)L)$, $h(M \oplus L)$, $h(L+M \oplus L)$, ...,$h((Q-1)L+M \oplus L)$, ...,$h(((L-1)M) \oplus L)$, $h(L+((L-1)M) \oplus L)$, ..., $h((Q-1)L+((L-1)M) \oplus L)$, then it can be addressed sequentially for Q of its values for computing each output sample.

V. Present Work

In the present work, both non-optimal (single-stage) and optimal (multi-stage) interpolators and decimators have been implemented for interpolation ratios of 3/2, 3 and 5 to get at sampling rates of 10 KHz, 20 KHz, and 33 KHz, starting from 6.67 KHz, and decimation ratios of 3/2, 3 and 5 to get back to sampling rates of 6.67 KHz.

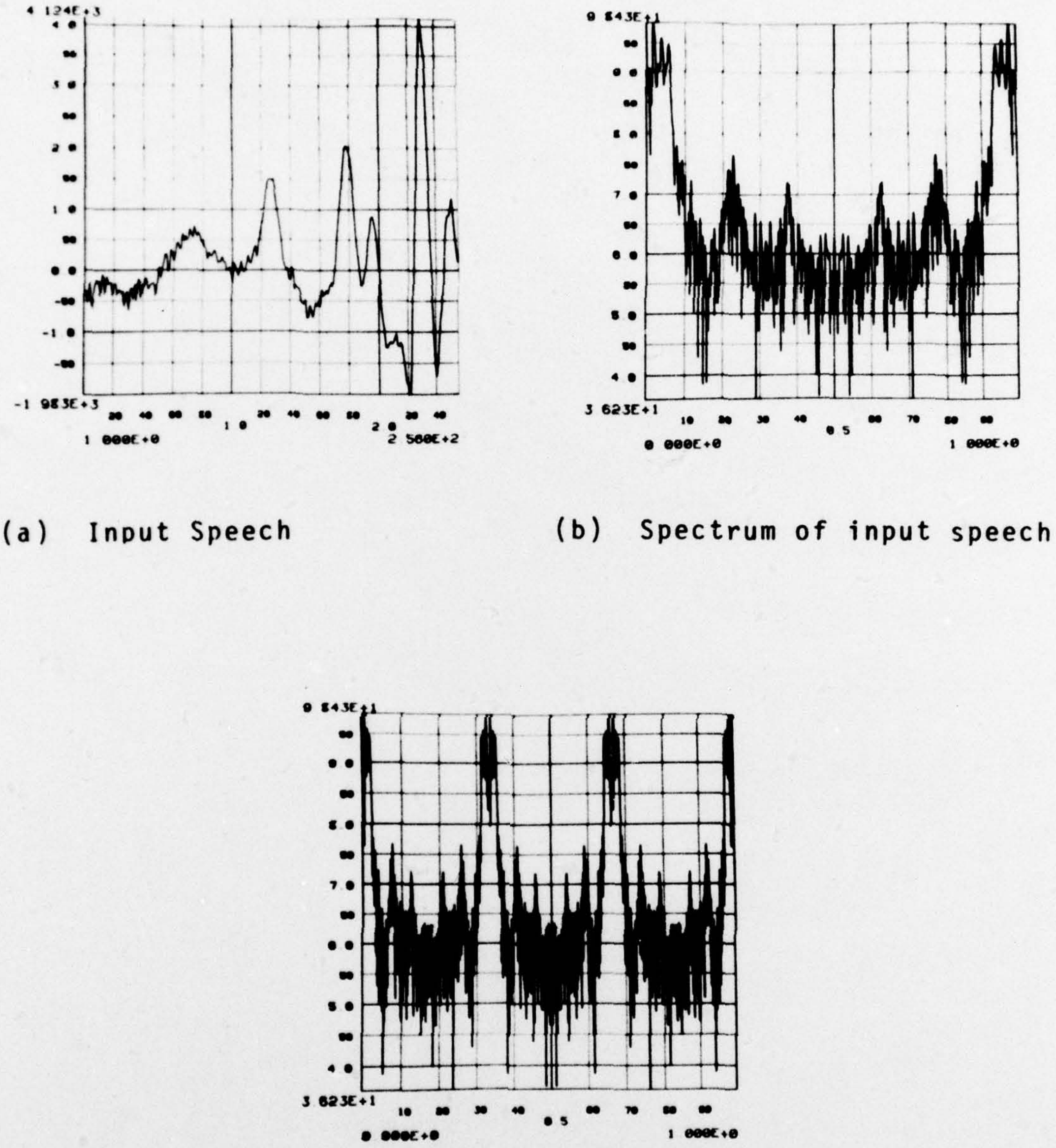
From the design curves, it was determined that two stage implementation was optimal for ratios of 3 and 5, and one stage for a ratio of 3/2. Optimal values for the interpolation (decimation) ratios for both the stages are calculated and filters specified. The filters were designed using the REMEZ exchange algorithm. In the multistage implementation, the filters are scrambled as explained in the previous section and stored on files. The two implementations were tested for run times and results follow.

Integer ratio	Single stage interpolator	Multistage interpolator
3/2	70 seconds	20 seconds
3	72 seconds	29 seconds
5	150 seconds	34 seconds

The spectra of original speech and interpolated speech signals are presented in Figure 7 which clearly show the frequency domain effects of interpolation. The language used for implementation is FORTRAN. (Listings of single-stage interpolator and multi-stage interpolator and decimators are enclosed.)

VI. Suggestions for Improvement

Goodman and Carey propose a set of nine FIR filters of which eight are halfband filters, to be used in proper sequence, along with resampling to real'ze a wide range of accurate interpolation and decimation ratios. It is claimed that the filters are efficient in terms of number of multiplications, since most of the filters in the set are half band filters (in which nearly half of the impulse response coefficients are zero) and that the required filter sequence can be designed without a computer.

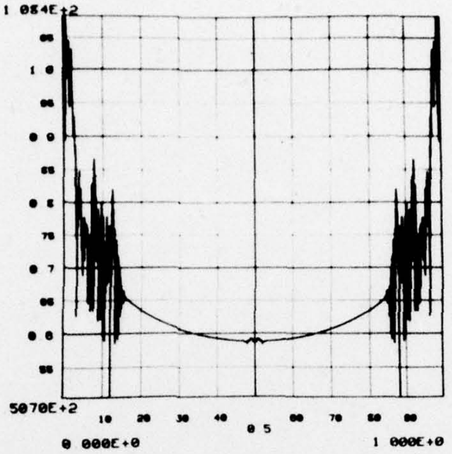


(a) Input Speech

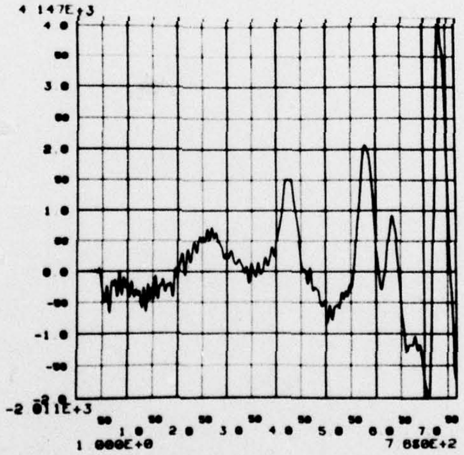
(b) Spectrum of input speech

(c) Spectrum of interpolated but unfiltered speech

Figure 7: Input and output speech signals and their spectra (Interpolation ratio = 3/1)



(d) Spectrum of optimally interpolated speech (D = 3/1)



(e) Optimally interpolated speech (D = 3/1)

Figure 7 (continued)

SECTION V

CONTINUOUSLY VARIABLE SLOPE DELTA MODULATOR

H. Ravindra

I. Introduction

Delta modulation is a waveform encoding technique which is characterized by one bit information over each clock period. In its simplest form, the modulator consists of a comparator (Ref. Figure 1), sampler and an accumulator. The comparator compares the input signal and the accumalator contents, and produces an error signal at the data sampling rate. The sampler generates a one bit output 1 or 0 depending on whether the error signal is positive or negative respectively. A fixed step, Δ, is either added to or subtracted from the accumulator when the output bit is a 1 or 0 respectively - then the accumulator content will be an approximation to the original speech signal. So, the bit stream output by the encoder represents an approximation for the original signal. The decoder is similar to the encoder except for the absence of the comparator and the presence of a LP filter at the output. It has been shown that a simple scheme like this suffers from two main disadvantages. They are:

(i) slope overload noise and

(ii) granular noise.

Slope overload results if the input speech signal to the encoder varies so fast that the encoder cannot track it well and granular noise results due to the alternation of 1 and 0 to represent a constant level. These noises can be reduced considerably by adapting the step size dynamically so that when the slope is high, the step size increases and when the slope is very low, the step size decreases. In the digital adaptive delta modulators, the present and previous bits are compared and if they are the same (indicating slope overload), the step size is increased and if they are opposite, the step size is reduced. In a CVSD encoder, the step size is adapted more smoothly in time, with a time constant that is of the order of 5 - 10 ms. to match the syllabic rate of speech. It has been claimed that CVSD processed speech sounds "cleaner" at bit rates of 25 K bits/second and lower. Also, this method provides for an increased resistance to bit errors. But the price paid is increased slope overload distortion compared with instantaneous compandors.

II. CVSD Encoder and Decoder [Reference: Figure 1]

The comparator compares the audio input with the output of the integrator and produces an error signal. The sampler produces a 1 or 0 depending on whether an error signal is positive or negative respectively at that clock instant. The algorithm detects three like consecutive bits and outputs a '1' and a '0' otherwise. The algorithm output is at clock rate. The slope command low pass filters the algorithm output and produces a signal which acts as the modulating signal in the PAM. The slope command LP filter is selected such that its time constant matches the syllabic rate of speech. As can be seen, the modulating signal increases exponentially if there is a slope overload, and hence results in increasd step size. The PAM modulates the sampler output with the slope command output. The integrator constructs an approximation for the original audio signal. The decoder is similar to the encoder except for the absence of the comparator. It is suggested that the slope command filter should have a pass band width of 25 to 40 Hz and that the compression ratio should be 12:1.

III. Present Implementation

The CVSD system is simulated on a digital computer. The algorithm is implemented by a 3 bit Shift Register which stores the present bit and the past two bits and produces an output of '1' if all the bits are alike and '0' otherwise. The slope command filter is a digital LP filter with a passband cutoff between 25 and 40 Hz and a roll off rate of 20dB per decade. The slope command filter is convolved with the algorithm output to generate the slope command signal. This signal modulates the sampler output and is summed up with the proper sign in the accumulator, and then low pass filtered to generate an approximation for the original audio signal. Since this is a feedback system, it is necessary to generate outputs from the slope command module, the PAM and the Integrator one point at a time. So, the convolutions mentioned earlier are carried out such that one point is generated and output from the module at a time. Two operations are to be performed on the slope command output before it is sent to the PAM. They are,

(i) Biasing: When the speech input is zero, the sampler output will be a sequence of alternating 1's and 0's. So, the modulating signal will be zero and this requires the addition of a bias to the slope command signal so that the approximation to a constant level is a train of alternating 1's and 0's of small magnitude.

(ii) Amplification: When the slope of the input signal is large over a long period of time, the slope command output saturates at the maximum possible value. This value may not be sufficient to provide for closer tracking of the high slope signal. Therefore the slope command output must be amplified.

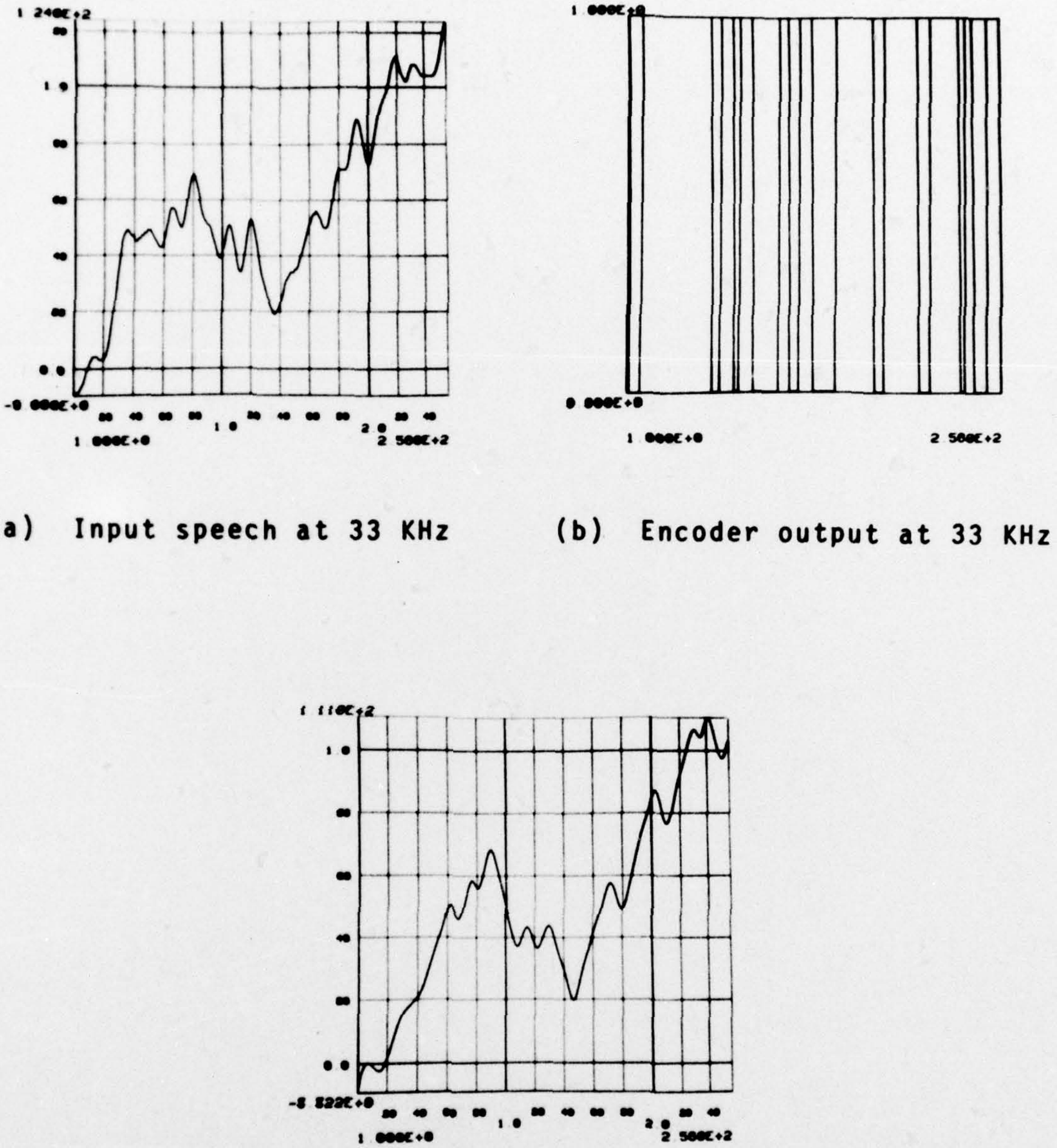
The biasing and amplification can be adjusted so that the compression ratio is 12:1 as specified.

The FIR filters are designed using the REMEZ exchange algorithm. The system reads in speech from a file, encodes it and passes the encoded output into the decoder which decodes the encoded speech and produces the output speech which is written onto a file.

Results:

At this stage, the encoder and decoder are logically working correctly. It looks as if it is necessary to design and use separate slope command filters for the three possible sampling rates. Each filter can then be tuned up so that it matches the syllabic rate of speech and also the slope command output can be properly biased and amplified. The observation with a single filter designed for a passband cut off at 32 Hz based on a sampling rate of 20 KHz leads to heavy slope overload when the sampling rate is lower. The integrator filter should be such that the base band of the reconstructed speech passes through.

The output of each module is presented in Figure 3.



(a) Input speech at 33 KHz

(b) Encoder output at 33 KHz

(c) Decoder output at 33 Khz

Figure 3: Input and outputs of the CVSD System